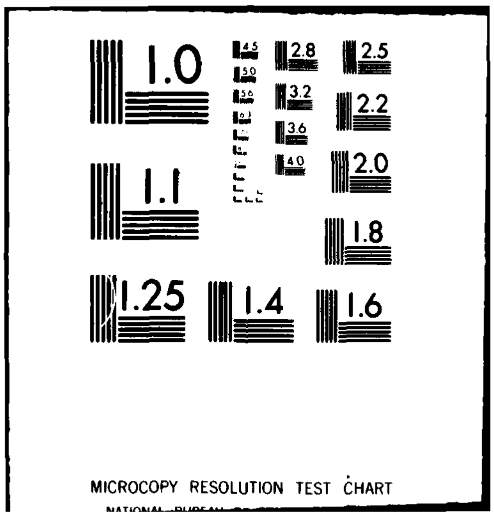
AD-A090 966 TRW DEFENSE AND SPACE SYSTEMS GROUP REDONDO BEACH CA F/G 9/2
AIRBORNE SYSTEMS SOFTWARE ACQUISITION ENGINEERING GUIDEBOOK FOR--ETC(U)
SEP 78 D J REIFER F33657-76-C-0677
UNCLASSIFIED TRW-30323-6009-TU-00 ASD-TR-79-5028 NL

1 of 2
AD A
090966


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART

LEVEL II

ASD-TR-79-5028

4w

1

Airborne Systems

Software Acquisition Engineering Guidebook

for

VERIFICATION, VALIDATION AND CERTIFICATION

AD A090966

SEPTEMBER 1978

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED

PREPARED FOR
DEPUTY FOR ENGINEERING
AERONAUTICAL SYSTEMS DIVISION
WRIGHT-PATTERSON AFB, OH 45433

PREPARED BY
TRW DEFENSE AND SPACE SYSTEMS GROUP
ONE SPACE PARK
REDONDO BEACH, CA 90278

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

JOHN M. HOEFERLIN, Project Engineer
Information Engineering Division

RICHARD J. SYLVESTER
ASD Computer Resources Focal Point

FOR THE COMMANDER

ROBERT P. LAVOIE, Colonel, USAF
Director of Avionics Engineering
Deputy for Engineering

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify ASD/ENAIA, W-PAFB, OH 45433 to help us maintain a current mailing list".

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ASD-TR-79-5028	2. GOVT ACCESSION NO.: AD-A090 966
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): Airborne Systems Software Acquisition Engineering Guidebook for Verification, Validation, and Certification	5. TYPE OF REPORT & PERIOD COVERED: FINAL rept.
	6. PERFORMING ORG. REPORT NUMBER: TRW-30323-6009-TU-00
7. AUTHOR(s): D. J. Reifer	8. CONTRACT OR GRANT NUMBER(s): F33657-76-C-0677
9. PERFORMING ORGANIZATION NAME AND ADDRESS: TRW Defense and Space Systems Group, One Space Park, Redondo Beach, CA 92078	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: PE64740F Project 2238
11. CONTROLLING OFFICE NAME AND ADDRESS: Hq ASD/ENAIA, Wright-Patterson AFB OH 45433	12. REPORT DATE: September 1978
	13. NUMBER OF PAGES: 120
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Software VV&C, Independent Verification and Validation, V & V Tools

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report is one of a series of guidebooks which provide guidance to the acquisition management and engineering of Airborne Systems software procured under Air Force 800-series regulations. It provides working-level Air Force Program Office engineering and management personnel with information that will help them plan, specify, and monitor independent computer program Verification, Validation and Certification (VV&C) activities.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED 409637

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PREFACE

This guidebook is one of a series of guidebooks intended to assist Air Force Program Office and Engineering personnel in software acquisition engineering for airborne systems. The contents of the guidebooks will be revised periodically to reflect changes in software acquisition policies and practices and feedback from users.

This guidebook was prepared under the direction of the Aeronautical Systems Division, Deputy for Engineering (ASD/EN) in coordination with the Space Division, Deputy for Acquisition Management (SD/AQM).

The entire series of Software Acquisition Engineering Guidebooks (Airborne Systems) is listed below along with ASD Technical Report numbers and NTIS accession numbers where available.

Regulations, Specifications and Standards	ASD-TR-78-6	ADA058428
Reviews and Audits	ASD-TR-78-7	ADA058429
Software Quality Assurance	ASD-TR-78-8	ADA059068
Configuration Management	ASD-TR-79-5024	ADA076542
Computer Program Documentation Requirements	ASD-TR-79-5025	ADA076543
Statements of Work and Requests for Proposal	ASD-TR-79-5026	ADA076544
Requirements Analysis and Specification	ASD-TR-79-5027	
Verification, Validation and Certification	ASD-TR-79-5028	
Microprocessors and Firmware	ASD-TR-80-5021	
Software Development Planning and Control	ASD-TR-80-5022	
Software Testing and Evaluation	ASD-TR-80-5023	
* Contracting for Software Acquisition	ASD-TR-80-5024	

* Software Cost Analysis and Estimating	ASD-TR-80-5025
* Supportable Airborne Software	ASD-TR-80-5026
* Software Development and Support Facilities	ASD-TR-80-5027
* SAE Guidebooks - Application and Use	ASD-TR-80-5028

* These Guidebooks Available Fall 1980.

CONTENTS

CONTENTS (Continued)

CONTENTS (Continued)

TABLES

TABLES (Concluded)

ILLUSTRATIONS

ABBREVIATIONS AND ACRONYMS

AF	Air Force
AFR	Air Force Regulation
AFSC	Air Force Systems Command
ASD	Aeronautical Systems Division
ASPR	Armed Services Procurement Regulation
CDR	Critical Design Review
CDRL	Contract Data Requirements List
CI	Configuration Item
CMP	Configuration Management Plan
CPAF	Cost Plus Award Fee
CPC	Computer Program Component
CPCI	Computer Program Configuration Item
CPDP	Computer Program Development Plan
CPIF	Cost Plus Incentive Fee
CPQAP	Computer Program Quality Assurance Plan
CRISP	Computer Resources Integrated Support Plan
CRWG	Computer Resources Working Group
DID	Data Item Description
DOD	Department of Defense
DODD	Department of Defense Directive
DODI	Department of Defense Instruction
DSARC	Defense Systems Acquisition Review Council
DR	Discrepancy Report
DT&E	Development Test and Evaluation

ABBREVIATIONS AND ACRONYMS (Continued)

EW	Electronic Warfare
FCA	Functional Configuration Audit
FPI	Fixed Price Incentive
FQR	Functional Qualification Review
FQT	Final Qualification Test
GFD	Government Furnished Data
GFE	Government Furnished Equipment
GO	Government Organic
HQ	Headquarters
ICD	Interface Control Document
ICT	Independent Contractor Team
IFB	Information for Bidders
IOT&E	Initial Operational Test and Evaluation
ITG	Independent Test Group
IV&V	Independent Verification and Validation
LCC	Life Cycle Cost
LOE	Level of Effort
MIL STD	Military Standard
MM	Minuteman
MOA	Memorandum of Agreement
NSCCA	Nuclear Safety Cross Check Analysis
OI	Operating Instruction
OSD	Office of the Secretary of Defense
OT&E	Operational Test and Evaluation
PCA	Physical Configuration Audit

ABBREVIATIONS AND ACRONYMS (Continued)

PDR	Preliminary Design Review
PM	Program Manager
PMP	Program Management Plan
PMRT	Program Management Responsibility Turnover
PO	Program Office
PQT	Preliminary Qualification Test
QA	Quality Assurance
RFP	Request for Proposal
ROI	Return on Investment
SAEG	Software Acquisition Engineering Guidebook
SAMSO	Space and Missile Systems Organization
SDR	System Design Review
SEMP	System Engineering Management Plan
SON	Statement of Operational Need
SOW	Statement of Work
SPO	System Program Office
SRR	System Requirements Review
T&E	Test and Evaluation
TEMP	Test and Evaluation Master Plan
TEOA	Test and Evaluation Objectives Annex
TRR	Test Readiness Review
USAF	United States Air Force
V&V	Verification and Validation
V, V&C	Verification, Validation and Certification
VVD	Verification and Validation Director
VVMP	Verification and Validation Master Plan
WBS	Work Breakdown Structure

1. INTRODUCTION

The purpose of this guidebook is to provide working-level Air Force Program Office engineering and management personnel with information that will help them plan, specify, and monitor independent computer program Verification, Validation and Certification (V, V&C) activities in connection with the acquisition of Computer Program Configuration Items (CPCI's) for airborne systems.

The guidelines, checklists and references in this guidebook serve to supplement the guidance in Air Force Regulation (AFR) 800-14, Volume II, "Acquisition and Support Procedures for Computer Resources in Systems," and AFR 80-14, "Test and Evaluation", which are the primary documents governing computer program Verification, Validation and Certification activities within the software acquisition process.

1.1 PURPOSE OF VERIFICATION, VALIDATION AND CERTIFICATION

The purpose of Verification, Validation and Certification is to provide the Air Force Program Office (PO) with systematic assurance that the computer programs they acquire will perform their mission requirements economically, efficiently and correctly. This assurance is enhanced by having an objective third-party independently assess the technical adequacy of the delivered software products.

The concept of Verification, Validation and Certification was first employed in early space and missile systems where the consequences of failures were often catastrophic. The concept was extended to encompass nuclear safety analysis and command and control systems and is currently being used on a wide range of systems. Although a quantitative measure of the effectiveness of its application is impossible to make, an examination of the success of past systems employing it indicates that its added expense is justified. The following examples illustrate this point.

The Space and Missile Test Center's verification and validation contractor was tasked to independently evaluate and test a 25,000 word program that had been an integral part of the range safety system at Vandenberg AFB for the previous eight years. Twenty major errors were detected, seven of which were critical to range operation. Possible injury to life and/or property could have occurred if these errors were left uncorrected.

The Minuteman Program Office has employed an independent contractor to do Verification and Validation† for many years. Their overall error history illustrates the benefits attributed to this practice. Minuteman has experienced 1 error per 6000 lines versus an industry average of 1 error per 300 lines. This represents a 20 to 1 improvement.

Other projects such as the Titan missile system, the B-1 Bomber and the Safeguard anti-ballistic missile system have reported like successes. The success of the approach is epitomized by SAMSO Commander's Policy which directs that Independent Verification and Validation will be considered for all space and missile programs employing embedded computer resources.

Verification, Validation and Certification embody a series of activities which are ideally interfaced with the development process itself. The activities accomplished result in a more orderly and efficient implementation because each development phase produces a verified baseline for the next phase. In addition, errors are typically found early in the cycle before they have a chance to propagate. In summary, the three major payoffs of Verification, Validation and Certification are:

- Improved reliability - fewer errors after acceptance
- Greater visibility - improved chances of success
- Reduced life cost - errors found earlier

† Currently designated as Performance Analysis and Technical Evaluation (PATE).

1.2 VERIFICATION, VALIDATION AND CERTIFICATION DEFINED

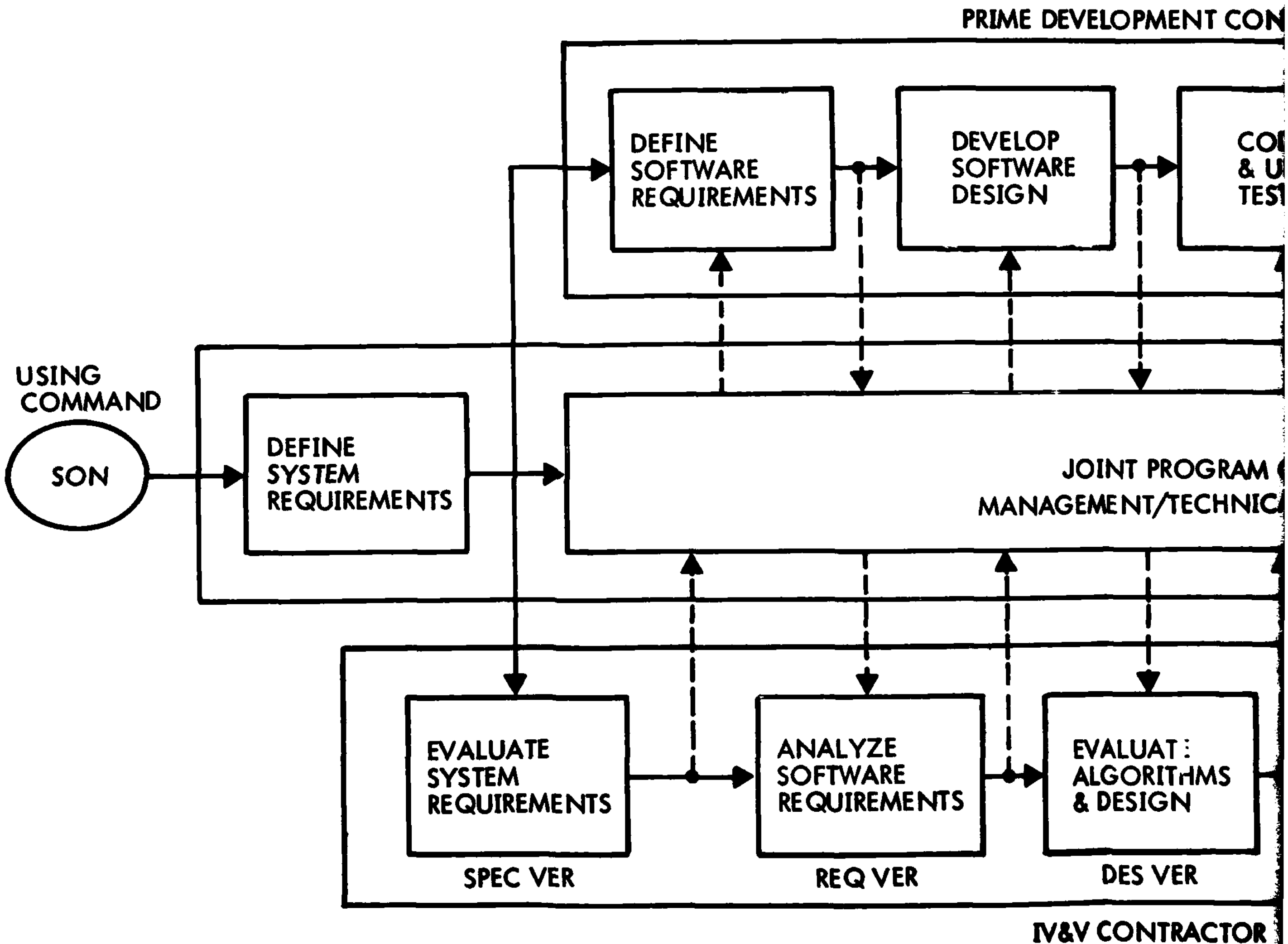
The terms Verification, Validation and Certification are being used extensively and somewhat interchangeably by members of the software community to describe many disparate testing and analysis activities. Service memos and regulations are often vague and conflicting when discussing the subject matter. The dictionary offers little relief from the confusion because the terms are synonyms for one another. Just what do the terms mean and what activities do they encompass?

Figure 1-1 illustrates Verification, Validation and Certification activities within the context of the sequential life cycle where software is acquired contractually. The purpose of so ordering the development is to create a series of validated baselines upon which the software products can be developed and tested. Typically, these baselines are documents which specify either requirements or programs as actually built. As can be seen in the figure, Verification, Validation and Certification provide management with the feedback they need to manage effectively. The following subparagraphs define the terms Verification, Validation and Certification within the context of this acquisition life cycle model. The subparagraphs defining the terms also describe each of the activities illustrated in the figure.

Table 1-1 summarizes what Verification, Validation and Certification is and is not. It is provided to clarify any misconceptions about the processes. For the purpose of this guidebook, verification and validation are conducted by personnel who are not associated with the development organization. This is the key discriminator between Verification and Validation (V&V) and Development Test and Evaluation (DT&E). V&V is sometimes defined to encompass Nuclear Safety Cross Check Analysis (NSCCA). Validation is sometimes defined to encompass Operational Test and Evaluation (OT&E). This guidebook does not address either of these two subjects.

1.2.1 Verification

Verification, as used in this guidebook series, is the iterative process of determining whether the product of each step of the Computer


PRIME DEVELOPMENT CON
DEFINE SOFTWARE REQUIREMENTS
DEVELOP SOFTWARE DESIGN
USING COMMAND
SON
DEFINE SYSTEM REQUIREMENTS
JOINT PROGRAM
MANAGEMENT/TECHNIC
EVALUATE SYSTEM REQUIREMENTS
ANALYZE SOFTWARE REQUIREMENTS
ALGORITHMS & DESIGN
SPEC VER
REQ VER
DES VER
IV&V CONTRACTOR

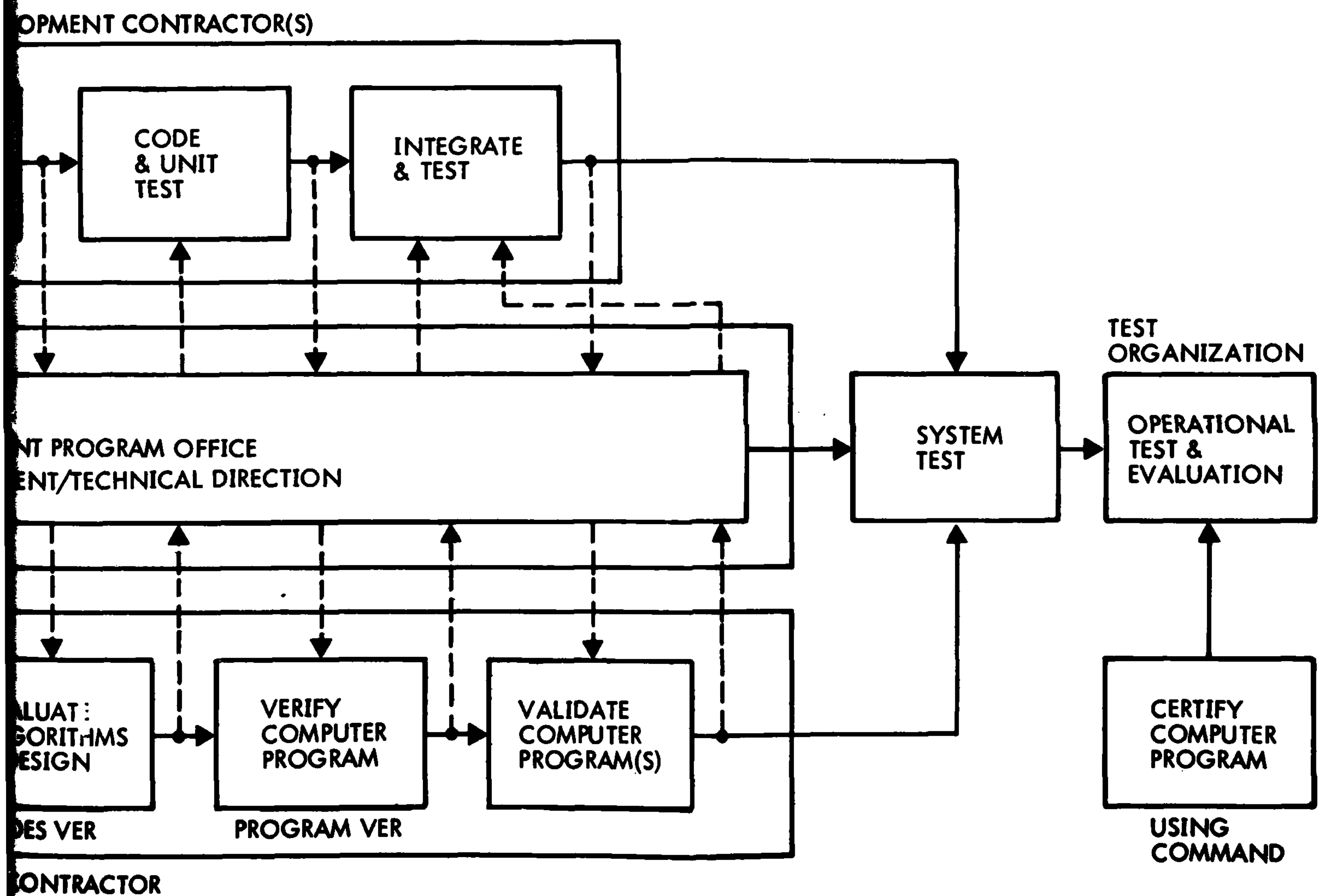


Figure 1-1. Acquisition Life Cycle Model

Table 1-1. Verification, Validation and Certification Explained

Independent Verification and Validation Is	Independent Verification and Validation Is Not
• An independent technical activity. • Aimed at product evaluation throughout the life cycle. • Identifying errors early. • Employed to insure that all system and subsystem requirements have been fulfilled by the software. • Complementary to the development effort. • Designed to help the developer. • Additional insurance.	• Conducted by the personnel that develop the software. • Checking the code during Development Test and Evaluation (DT&E). • Identifying errors during DT&E. • Employed to insure that only the test requirements of the computer program development specification are met. • A duplication of development activities. • Conducted to harass the developer. • A guarantee of success.
Certification Is	**Certification Is Not**
• An administrative process leading to approval. • Retrospective - it insures that everything required has been completed.	• A set of operational tests that are conducted to approve a product.

Program Configuration Item (CPCI) development and change process fulfills the requirements levied by the previous step. This definition is fully compatible with that contained within AFR 800-14, Volume I/ AFSC Supplement 1.

The four activities that comprise the verification process are briefly described as follows:

- System Specification Verification

  The system-level analytical activity conducted to determine whether the computer-applicable requirements within the System Specification represent a clear and accurate translation of the user's need as stated in his Statement of Operational Need (SON)† document.

- Requirements Verification

  The data system analysis (i. e., hardware and software) activity conducted to determine whether the software requirements (as specified Part I Development Specification) reflect the computer-applicable needs denoted by the System or System Segment Specification.

- Design Verification

  The software design analysis activity conducted to determine whether the software design represents a clear, consistent and correct mechanization of the requirements contained within the Part I Development Specification.

- Program Verification

  The code analysis and test activity conducted to determine whether the actual code correctly implements the design as described in its associated documentation and whether it is compliant with the Draft Part II Product Specification which contains the design.

1.2.2 Validation

Validation, as used in this guidebook, encompasses the evaluation, integration and test activities conducted at the system level to ensure that the finally developed software satisfies applicable requirements set down as performance and design criteria in the System or System

† Formerly designated as a Required Operational Capability (ROC).

Segment Specification and/or the Part I Development Specification. This definition is fully compatible with that contained within AFR 800-14, Volume I/AFSC Supplement 1.

Successful validation requires that all verification activities are completed. This is necessary because verification procedures often provide a basis for selection of the validation approach.

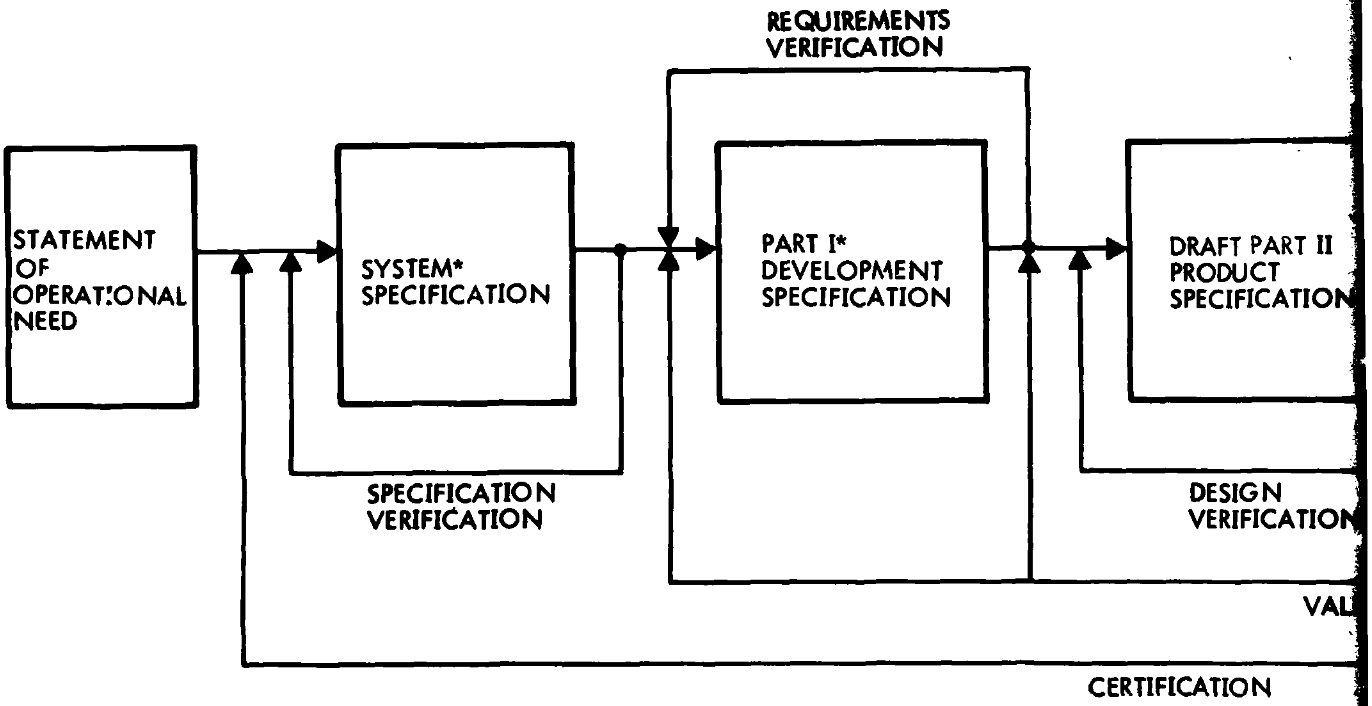
Validation is usually conducted to ensure system-level requirements are fulfilled. Therefore, software's contribution to performance must be evaluated in a realistic operating environment where hardware, environmental and personnel effects are in the loop.

1.2.3 Certification

Certification, as used in this guidebook, refers to the formal administrative procedures established to substantiate that enough evidence has been obtained to state with near certainty that the acquired system and its attendant software's performance will satisfy the user's documented need. As such, certification embodies all the test and evaluation and verification and validation activities conducted during the Development Test and Evaluation (DT&E) phase and includes Initial Operational Test and Evaluation (IOT&E).

1.3 LIFE CYCLE RELATIONSHIPS

The primary function of Independent Verification and Validation is to provide the PO with assurance that his software products will satisfy the user's intended need in a cost and schedule effective manner. This function is accomplished in a series of steps which are ideally interfaced through the PO to the development activity itself. In this manner, each development phase provides a definitive, verified baseline prior to the initiation of the next phase. Figure 1-2 illustrates these relationships pictorially. These relationships are fully compatible with those displayed in Figure 1-1 where the products of these phases are analyzed.



*INCLUDES ASSOCIATED INTERFACE CONTROL DOCUMENTS.

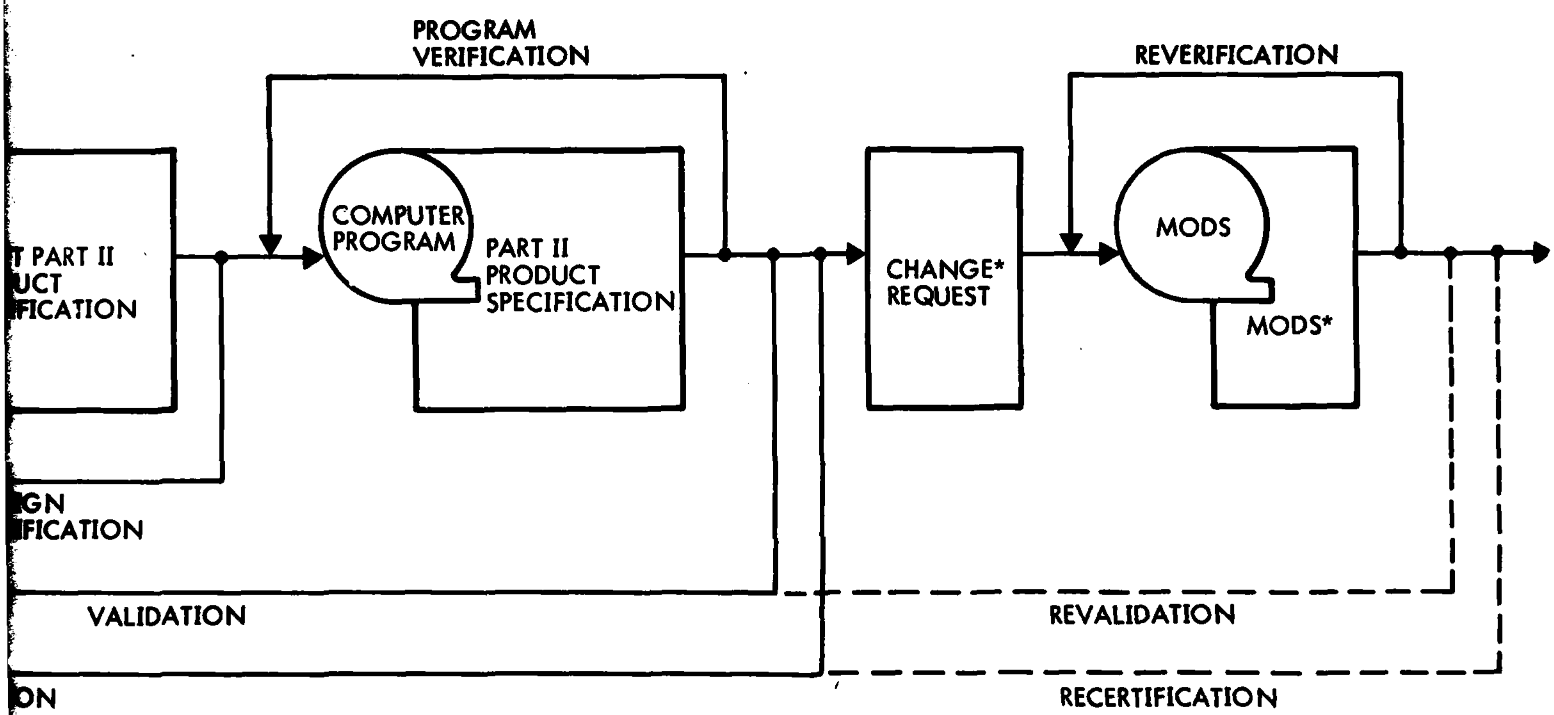


Figure 1-2. Relationship of IV&V Development Process

1.4 RELATIONSHIP TO OTHER GUIDEBOOKS

This guidebook is one of a series of interrelated volumes that can be used effectively together to acquire avionics system software. It draws from the knowledge reported in the other volumes in the series and incorporates key concepts. For example, all of the guidebooks apply when the PO elects to acquire the services of an IV&V contractor as described within this volume. They all apply because software and software-related services are being acquired contractually by the government.

1.5 CONTENTS OF THE GUIDEBOOK

1.5.1 Chapter 1: Introduction

Provides background information, defines the concepts, relates the concepts to the system acquisition life cycle and shows the relationship between this and other Software Acquisition Engineering Guidebooks (SAEG's).

1.5.2 Chapter 2: Relevant Documents

References the government regulations, specifications and standards relevant to Verification, Validation and Certification.

1.5.3 Chapter 3: General Guidelines for Independent Verification and Validation

Provides general guidance to the PO and engineering personnel in planning, organizing, staffing and control of IV&V activities.

1.5.4 Chapter 4: Specific Guidance for the Conduct of Verification, Validation and Certification

Provides detailed guidance for the conduct of Verification, Validation and Certification activities. For each activity, it identifies the responsibilities of the participating organizations and discusses applicable concepts, methods, products, and problems.

1.5.5 Appendix A: Glossary of Key Terms

Provides definitions for the major terms used within this guidebook.

1.5.6 Appendix B: Tools and Techniques Survey

Provides a listing of available Verification, Validation and Certification tools and techniques by activity.

1.5.7 Appendix C: Bibliography

Lists a number of references that provide insight into various aspects of Verification, Validation and Certification.

2. RELEVANT DOCUMENTS

2.1 REGULATIONS, SPECIFICATIONS, AND STANDARDS

The following government documents are important sources of information relevant to Verification, Validation and Certification of computer programs:

- Directives

DODD 5000.3	Test and Evaluation
DODD 5000.29	Management of Computer Resources in Major Defense Systems

- Military Standards

MIL-STD 483 (USAF)* 31 December 1970	Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs
MIL-STD 490 30 October 1968 Notice 2 18 May 1972	Specification Practices
MIL-STD 1521A (USAF) 1 June 1976	Technical Reviews and Audits for Systems, Equipment, and Computer Programs

- Air Force Documents

AFR 80-14 19 July 1976	Test and Evaluation
AFR 122-9 1 July 1974	The Nuclear Safety Cross-Check Analysis and Certification Program for Weapon Systems Software
AFR 122-10 7 November 1975	Nuclear Weapon Systems Safety Design and Evaluation Criteria
AFR 800-14, Volume I 12 September 1975 AFSC Supplement 1 8 August 1977	Management of Computer Resources in Systems

*Currently being revised.

AFR 800-14, Volume II 26 September 1975	Acquisition and Support Procedures for Computer Resources in Systems
SAMSOR 5-4 Attachment 9 5 January 1978	Commander's Policy, Management of Computer Resources in SAMSO Programs/Projects
MN OI 122-2 22 April 1977	Nuclear Safety Cross-Check Analysis

2.2 OTHER DOCUMENTS

A bibliography of relevant literature, including reports produced under government contract, is contained in Appendix C of this guidebook.

3. GENERAL GUIDELINES FOR INDEPENDENT VERIFICATION AND VALIDATION

This section provides general guidance in planning, organizing, staffing, directing, and controlling an Independent Verification and Validation (IV&V) effort.

3.1 RESPONSIBILITIES

Independent Verification and Validation represents a practical methodology that can be employed by Air Force PO personnel to cope with the problems associated with acquiring computer-based systems. Effective utilization of the methodology is achieved when the PO dedicates the appropriate resources (manpower and dollars) necessary to fulfill the following managerial responsibilities:

- Prepare a V&V Master Plan (VVMP) during the Conceptual Phase to be included as part of the PMP.
- Create the organizational focus within the PO to manage the contemplated IV&V effort.
- Staff the effort using either government or contractor personnel. If contractor personnel are used, the PO should prepare for and conduct a competitive source selection.
- Direct the effort toward achievement of its goals (stated in the VVMP).
- Control the effort to ensure acceptable technical, cost, and schedule performance.

Key questions that should be answered early in the planning stage include:

- Is IV&V necessary?
- How much IV&V is enough?
- When should IV&V start?

Once these questions are answered, the following decisions should be made:

- What is the proper organization to perform IV&V during each major life cycle stage?

- What IV&V tasks are necessary to support identified milestones and what are the deliverables?
- What unique personnel requirements do these tasks impose?
- How much money should be allocated to accomplish these tasks?

The PO can fulfill their responsibilities and answer these and similar questions by assigning an individual the task of preparing the VVMP. The importance of early planning cannot be over-emphasized. The U.S. General Accounting Office reviewed the problems associated with acquiring computer-related equipment and services and concluded that a majority of these could have been avoided if proper planning had taken place.

The person who is delegated the IV&V authority is referred to in this guidebook as the Verification and Validation Director (VVD). The VVD's reporting relationship within the PO is very important. He should not report to the same person in the PO to whom the development contractor reports (normally the engineering director). Rather, he should report to a parallel organization such as the integration or test divisions. This reporting relationship preserves objectivity and provides the PO Director with an independent check and balance system.

The remainder of this chapter assumes that the reader is the VVD. Key management factors which impact the IV&V effort are discussed in the following sections. These discussions are directed towards answering the questions posed above. The specifics of IV&V are discussed in Chapter A.

3.2 GENERAL PURPOSE OF INDEPENDENT VERIFICATION AND VALIDATION

Independent Verification and Validation activities are aimed at providing the PO with systematic assurance that the software will do what it is supposed to do and nothing more. This is accomplished by having an independent agency objectively critique the developer's products. IV&V concentrates on identifying requirements and design errors early

in the life cycle and verifies through independent test and evaluation that the software is mechanized properly later on. Its benefits include:

- Early identification of ambiguous, ill-defined and inadequate software requirements
- Early and continued emphasis on test planning
- Detection and correction of improperly mechanized designs and code
- Improved PO visibility into the detailed status of the software development activity as it progresses
- Reduced incidences of software errors once the system is operational
- Ease of maintenance once the system is operational

The potential benefits of IV&V are not free. Experience indicates that IV&V activities cost from 20 to 60 percent of that expended for software development. Because of the cost, an IV&V activity should be initiated only when it is economically justified in terms of life cycle benefits. Examples of candidate systems for IV&V are as follows:

- Software with a high cost of failure (e.g., space systems).
- Software for which the cost of error detection through operational use is greater than the cost of IV&V (e.g., aircraft operational flight programs).
- Real-time software which must work under all scenarios (e.g., range or nuclear safety programs).

Factors used to evaluate the applicability of IV&V are discussed in Section 3.4. Needless to say, IV&V should be applied prudently to areas considered critical enough to warrant the cost.

3.3 PERFORMER ORGANIZATION OPTIONS

At the PO's option, and potentially changing during various phases of the life cycle, the V&V function can be performed by any or all of the following:

- An independent test group
- The government organically
- An independent contractor team

All three of these options have been used successfully on avionics projects. A chart summarizing the advantages and disadvantages as well as the scope of the effort supported is included as Table 3-1. Each of the options are discussed in more detail in the following paragraphs.

3.3.1 Independent Test Group

The first performer option to be examined is the Independent Test Group (ITG). Basically, an ITG is an organizational entity created as part of the development contractor's team to be responsible for independently testing and evaluating the software development group's products. ITG characteristics are as follows:

- The ITG typically reports to the test or integration manager, while the development group reports to the engineering manager. This preserves objectivity.
- The ITG can be staffed internally or subcontracted. Either arrangement has proven satisfactory as long as the reporting independence is preserved.
- The ITG's job is test oriented. It does not purport to accomplish a full IV&V as defined in Chapter 4 above.
- The development contractor's program or facility manager is the final authority for settling disputes between the ITG and the development group.
- The ITG tends to use the same tools and facilities that the development group uses in their job. These could introduce the same errors and should be avoided.

Advantages and disadvantages of using an ITG are summarized in Table 3-2. As seen in the table, the main advantage of an ITG, its test orientation, is also a major disadvantage. The ITG is brought aboard specifically to test the developed software. They analyze the front-end products for testability, not correctness. Their primary goal is to find errors and to bring them to the attention of the developer for correction. The ITG assumes errors exist in the software and uses every trick of the trade to identify them during testing. Unfortunately, front-end analysis is sacrificed in this arrangement. Another major disadvantage of this approach is the questionable objectivity of the results. When development and testing are under the control of one contractor, pressures can be exerted to dilute or eliminate embarassing discoveries.

Table 3-1. IV&V Performer Alternatives

Alternative	V&V Scope Realized	Comments
Independent Test Group	• Independent Testing	• Available technical capabilities limited: best people usually assigned to development effort • Tool independency not maintained • PO visibility limited • Lowest cost • Questionable objectivity
Government Organic	• Independent Testing • Limited Systems Analysis and Audit Roles • Critical Element Analysis	• Available government resources limited • Tool independence maintained • Facilitates training and transition to O&M role • Improved management span of control

Independent Contractor Team	• Broad IV&V Capability
	  • Systems Engineering Support
	  • Independent Testing
	  • Audit Role
	  • Independent Analysis
		• Contractual commitment to do good job
		• Tool independence maintained
		• Provides second source for development
		• Highest cost
		• Objective and proven approach

Table 3-2. Independent Test Group Evaluation

Advantages	Disadvantages

• Test process oriented
  • Assures early test planning
  • Clearly defined test objectives
• Greater comprehensiveness in testing
• Experience continuity
  • Builds on accumulated test experience
  • Familiarity with use of test tools
• Least cost
	• Available technical capabilities limited
	  • Best people assigned to development effort
	• Tool independency not maintained
	• External visibility limited
	• Objectivity questionable
	• Lack of concentration on front-end

Guidelines for use of an ITG are provided in Table 3-3. Projects which are characterized by low to medium cost and risk seem to benefit most from an ITG approach.

3.3.2 Government Organic

The second performer option to be examined is Government Organic (GO). In this option, the government assumes the responsibility for conducting IV&V during either the early stages or during the operation and maintenance phase of the project. In either case, government personnel review the developer's products and provide feedback as to its correctness.

The GO approach offers the government many advantages. When properly conducted, it gets the user and supporting commands involved early. This early involvement fosters a team atmosphere. It also assists in providing essential user and supporter feedback to the developer through the PO for incorporation into his trade studies and requirements specifications. The GO approach can be characterized as follows:

- A government team of civil servants and military personnel assumes the responsibility for IV&V either during development or at the start of operations and maintenance.
- The government team reports to the VVD who had planned and who will direct their activities.
- The government team accomplishes the IV&V tasks called out in the VVMP according to agreed upon schedules using its own or a mix of government and contractor resources.
- The government team maintains facilities and tools that are independent of the development contractors. The tools are either developed by the government or transitioned for their use from an independent contractor source.

Advantages and disadvantages of using a GO approach are summarized in Table 3-4. The major advantage of this approach is the government's improved span of control. By conducting IV&V themselves, the government reduces the number of contracts it has to manage and the related administrative burden. It creates a larger and more talented team of technical and managerial specialists who will work with their contractors to ensure the technical adequacy of their products. Such teams are

Table 3-3. Performer Selection Guidelines

Factor	Independent Test Group	Government Organic	Independent Contractor Team
Type of Project	Consider for any avionics project.	Consider for large projects with a PMRT or small projects which are upgrades of current capabilities.	Consider for any avionics project.
Technical Risk	Consider when limited risk in requirements and limited exploitation of new technology.	Consider when there is large risk in requirements or schedules.	Consider when there is large risk in requirements and in implementation mistakes.
Criticality	Use when there is an identifiable critical function.	Use when identified critical functions and/or maintenance requirements justify IV&V.	Use for full range of criticality as justified by cost/benefit analysis.
Developer's Experience	Consider whenever developer has extensive experience in technical application area.	Consider whenever developer has limited experience in the user and supporter applications areas and these areas are essential for mission success.	Consider whenever developer lacks experience or is using new methods.
Available Dollars	Consider when there is a requirement for IV&V, but limited funds.	Consider after agreement is reached as to which agency will do what task and contribute what resources (dollars and people). Maintain reserves as contingency for contracting the IV&V in case of emergency.	Allocate sufficient funds to accomplish the tasks called out by the VVMP.
Available Personnel	Use when there are severe limitations in the number of PO personnel available.	Consider if the appropriate number of qualified and trained personnel can be made available.	Use if only a limited number of PO personnel available. Have them manage effort.

synergistic when adequately staffed. Unfortunately, acquiring the people to staff such teams is difficult. This is the major disadvantage of the approach.

Table 3-4. Government Organic Evaluation

Advantages	Disadvantages

• All of those for ITG (except cost)
• Improved management span of control
  • Fewer contracts to manage
  • Improved communications
• Facilitates training and transition to O&M role
• Objectivity least questionable
• Tool independence maintained
• Excellent visibility into critical areas
	• Available government resources limited
	  • Hard to get slots and good people to fill them
	  • People continuity difficult to maintain
	• Lead times for equipment acquisition can be prohibitive
	• Additional cost

Guidelines for use of a GO approach are provided in Table 3-3. Projects which are characterized by long life and frequent changes seem to benefit the most from a GO approach. Guidance in the area of scoping the IV&V activities to ascertain what level of IV&V is appropriate for a given project is presented in Paragraph 3.4.5 as are typical tasks and significant milestones.

3.3.3 Independent Contractor Team

The final performer option to be examined is the Independent Contractor Team (ICT). This option assumes that the government contracts with a contractor or contractor team to perform an IV&V of the development contractor's products.

The ICT is the most commonly used approach for IV&V. Its wide use is predicated on the fact that most PO's find it easier to get money allocated than people. The ICT approach can be characterized as follows:

- A contractor or contractor team is selected competitively to perform IV&V. A source slection is held and a contract is negotiated and signed.

- The ICT accomplishes the IV&V tasks called out in their contract according to an agreed upon schedule and an agreed to budget. The contractual IV&V tasks reflect those contained in the VVMP.

- The ICT reports to the VVD who has planned and who will technically direct their activities. The PO Director is the final authority for settling disputes between contractors.

- The ICT is systems engineering oriented. They concentrate on front-end analysis activities in hopes of identifying errors early in the development.

- The ICT develops and maintains facilities and tools that are independent of the development contractors. The tools can be transitioned to a GO team at the start of the maintenance phase, if desired.

Advantages of disadvantages of using an ICT approach are summarized in Table 3-5. The ICT represents a compromise which can

Table 3-5. Independent Contractor Team Evaluation

Advantages	Disadvantages

• All of those for ITG (except cost)
• Contractual commitment to do a good job
  • Ensures resource availability
  • Provides performance incentives
• Provides second source for development
• Objective and proven approach on large systems
• Tool independence maintained
• Excellent visibility into critical areas
	• Additional PO resources needed
	  • Source selection must be held
	  • Another contract must be managed
	• Potential communications problems and organizational conflicts
	• Large additional cost

help solve the PO's manpower problems. The major benefits of the GO approach can be captured at the cost of money and a limited number of PO personnel using the ICT approach. In addition, the ICT approach can set the stage for orderly transition to GO maintenance and provides a second source for development. The major disadvantage of the approach is its cost. However, this can be justified in many instances in terms of return on investment.

Guidelines for use of an ICT approach are provided in Table 3-3. Projects which are characterized by high technology and high cost of failure seem to benefit the most from the ICT approach.

3.4 PLANNING CONSIDERATIONS

This section discusses items that should be considered when formulating a VVMP. Each of the following subjects will be discussed in subsequent paragraphs:

- Tasks (Paragraph 3.4.1)
- Milestones (Paragraph 3.4.2)
- Deliverables (Paragraph 3.4.3)
- Contractual Provisions (Paragraph 3.4.4)
- Tailored Compliance Levels (Paragraph 3.4.5)
- Cost Estimating (Paragraph 3.4.6)

The discussion that follows is structured to assist the VVD in developing a meaningful IV&V program. Because of the wide variety of needs, there was no attempt made to provide a cookbook solution.

3.4.1 Typical Tasks

A list of representative IV&V tasks, major subtasks and appropriate organization(s) to perform them is illustrated in Table 3-6. Each task is briefly discussed in the following sub-paragraphs.

3.4.1.1 System Specification Verification

System specification verification is the V&V activity conducted prior to SDR to ensure that the system/system segment being considered

Table 3-6. Typical IV&V Tasks

Task	Subtasks	Performer
System Specification Verification	• V&V planning • Requirements analysis • Documentation review	• Government Organic (GO) • Independent Contractor Team (ICT)
Tool Development and Maintenance	• Tool evaluation • Tool development • Installation and demonstration • Training • Tool maintenance	• GO • ICT • Independent Test Group (ITG)
Software Requirements Verification	• Requirements analysis • Critical requirements identification • Documentation review	• GO • ICT
Software Design Verification	• Design analysis • Performance analysis • Documentation review	• GO • ICT
Program Verification	• Code analysis • Machine level testing • Documentation review	• GO • ICT • ITG
Software Validation	• Formal testing • DT&E review • Documentation review	• GO • ICT • ITG
Meeting Support	• Working groups • Reviews and audits • Management	• GO • ICT • ITG
Special Studies	• Quick-turn around studies • Design analysis trades	• ICT
Configuration and Data Management Support	• Configuration management • Data management	• ICT • GO

will fulfill its mission goals and objectives. The IV&V agency is typically brought on contract just after the PO approves the SRR minutes. They prepare a Verification and Validation Master Plan and initiate their tool development activity during this period. They review the validation phase products and actively participate in the SDR. A more detailed treatment of system specification verification is provided in Section 4.1.

3.4.1.2 Tool Development and Maintenance

As part of the V&V planning activity, the IV&V agency will identify an integrated set of tools for use throughout the life cycle. Some tools can be used as is or with minor adaptation. Others will have to be developed. The PO should coordinate with both the IV&V agency and the CRWG before initiating tool development. The IV&V agency will then develop these tools using a disciplined methodology which treats them as software products. If delivered, the completed tools should be installed and demonstrated prior to being accepted by the PO. User-oriented training should be provided to facilitate both the government's understanding of the tool's capabilities and/or the transfer of IV&V responsibility from a contractor to a government organization. Tool maintenance is provided as required. A more detailed coverage of tools is provided in Section 3.8. A glossary of available tools and techniques is included as Appendix B.

3.4.1.3 Software Requirements Verification

Software requirements verification is the V&V activity conducted prior to the end of the Validation Phase which ensures that the software requirements are an adequate translation of the system requirements allocated to software and that implementation is feasible. The IV&V agency evaluates the developer's products to ensure their technical adequacy and to identify those critical requirements for which the PO feels IV&V is economically justified. Requirements are analyzed and are sometimes independently derived in order to verify their viability. A more detailed treatment of software requirements verification is provided in Section 4.2.

3.4.1.4 Software Design Verification

Software design verification is the V&V activity conducted prior to CDR to ensure that both the software design represents a clear, consistent and accurate translation of the software requirements and the key algorithms perform with the required precision and accuracy. The IV&V agency evaluates the developer's products to ensure their technical adequacy and to contribute to the design refinement process. Key algorithms may either be simulated or rederived in order to demonstrate their technical viability. Timing and sizing budgets are monitored. A more detailed treatment of software design verification is provided in Section 4.3.

3.4.1.5 Program Verification

Program verification is the V&V activity conducted prior to the Final Qualification Test (FQT) to independently assure that the actual code developed is compliant with the approved design specification. The IV&V agency's responsibilities are to independently test and evaluate the developer's products using separate facilities and tools. The subject of program verification, how it differs from the developer's DT&E tasks and how it interfaces with validation are summarized in Section 4.4.

3.4.1.6 Software Validation

Software validation is the V&V activity conducted prior to system FCA to ensure that every requirement is adequately tested and that the software has been adequately shaken down from a system perspective. The IV&V agency tests and evaluates the code that was identified as critical in parallel with both its program verification and the developer's DT&E activities. A more detailed treatment of software validation is provided in Section 4.5. The discussion also treats the interfaces to the DT&E program.

3.4.1.7 Meeting Support

As part of mainstream V&V activities, the IV&V agency will participate in a number of meetings. These meetings include working groups established to get the IV&V and development agencies working together,

formal and informal (e.g., design inspections, code walk-thrus, etc.) reviews and audits, and a variety of project management meetings.

3.4.1.8 Special Studies

If desired, the PO can have the IV&V agency contractually support either quick-turnaround or design analysis trade studies. Quick-turnaround studies are typically conducted to work a specific problem area and recommend solutions. Design analysis trade studies are normally conducted to document the results of an important trade that impacts the software.

3.4.1.9 Configuration and Data Management Support

In a number of instances, non-critical configuration and data management tasks can be off-loaded to the IV&V agency. These activities would be in addition to those normally considered a part of the IV&V job.

3.4.2 Significant Milestones

Significant milestones associated with IV&V activities are shown in Figure 3-1. Phasing of each of the tasks noted in Paragraph 3.4.1 is illustrated in the figure.

3.4.3 Deliverables

A list of IV&V contract deliverables phased by task are provided in Table 3-7. The table also identifies appropriate Data Item Descriptions (DID's) and recommends a minimum set of data for an IV&V effort.

The list of deliverables may seem inordinately large. However, many of the documents are products of multiple tasks and do not represent a separate deliverable. For example, the software analysis report which is a product of the software requirements, design, and program verification tasks is one document updated during different phases of the life cycle. The reports appearing under different tasks with the same name should be counted only once.

It is important to note the need for interim delivery of the partially completed computer program(s). The IV&V agency cannot do their job if they have to wait for final products. They require timely access to the products as they are being developed. To provide interim deliverables, the development contractor must be tasked explicitly in his contract to

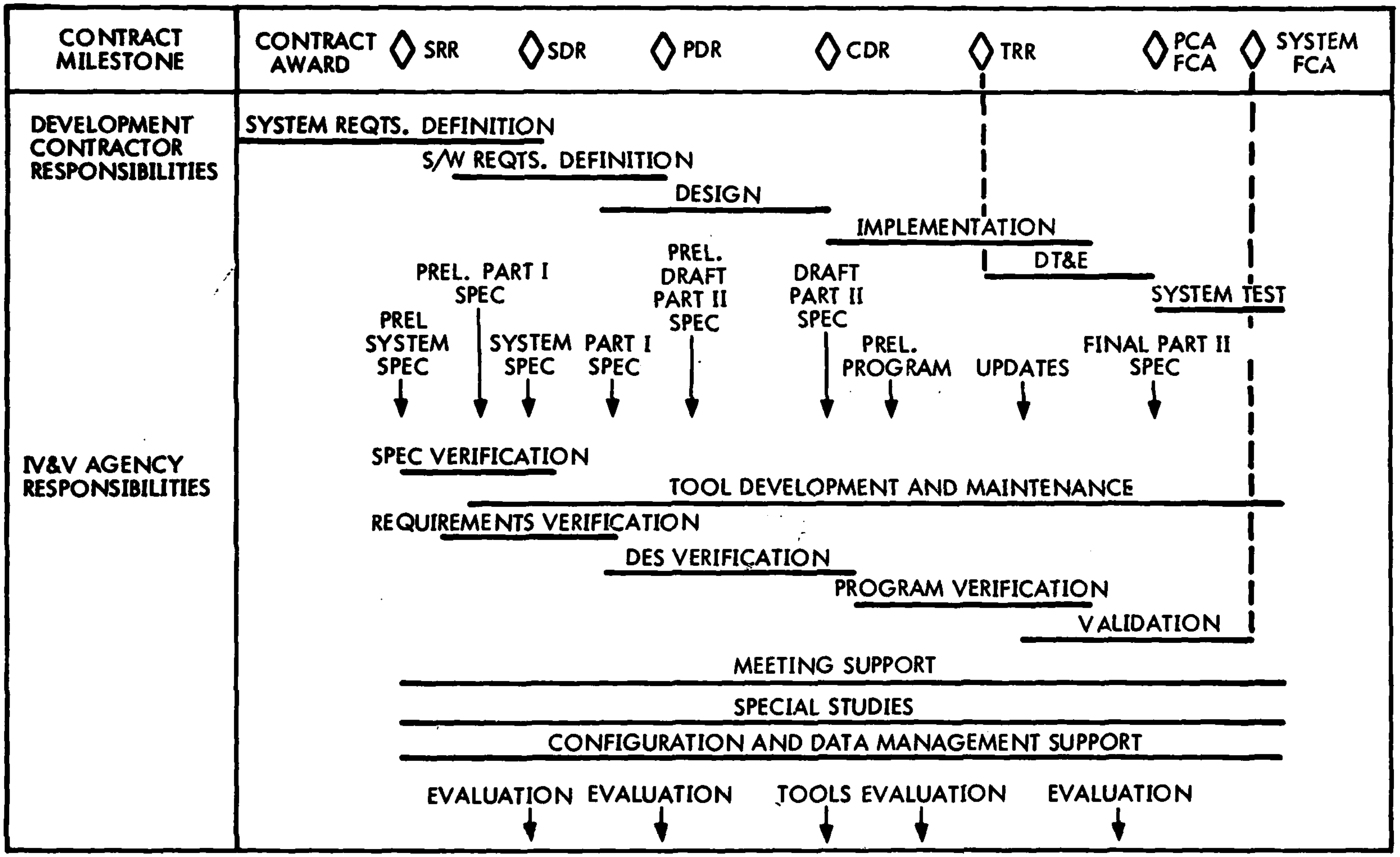


Figure 3-1. Significant IV&V Milestones and Task Phasing

Table 3-7. IV&V Deliverables

Task	Deliverables	DID Identifier
System Specification Verification	• Contractor V&V Master Plan†	‡
	• Configuration Management Plan†	E-3108
	• Software Analysis Report†	‡
Tool Development and Maintenance	• Tool Evaluation Report†	‡
	• Computer Program Development Plan	S-30567A
	• Part I Development Specification	E-3119A
	• Part II Product Specification	E-3120A
	• DT&E Test Plan/Procedures	T-3703
	• DT&E Test Report	T-3717
	• Installation Manual	‡
	• Users Manual	M-3410
	• Commercial Off-the-Shelf Manual†	M-7024
	• Version Description Document	E-3121
	• Training Support Data	H-3258A
	• Computer Program†	§
Software Requirements Verification	• Software Analysis Report†	‡
	◦ Requirements Analysis	
	◦ Timing and Sizing	
	• IV&V Test Plan/Procedures†	‡
Software Design Verification	• Software Analysis Report†	‡
	◦ Design Analysis	
	◦ Timing and Sizing	
	• IV&V Test Plan/Procedures†	‡
Program Verification	• Software Analysis Report†	‡
	◦ Code-Analysis	
	◦ Open-Loop Analysis	
	◦ Closed-Loop Analysis	
	• IV&V Test Report†	‡
	• IV&V Completion Letter†	‡
Software Validation	• Software Validation Report	‡
	◦ Test Evaluation	
	◦ Post Mortem	
	• IV&V Test Plan/Procedures†	‡
	• IV&V Test Report†	‡
IV&V Meeting Support	• Agenda-Design Reviews, Configuration Audits and Demos for IV&V Tools	A-3029
	• Minutes of Formal Reviews, Inspections and Audits for IV&V Tools	E-3118
	• Presentation Material†	A-3024
Special Studies	• Study Report	‡
	• Subsystem Design Analysis Report	S-3582
Configuration and Data Management Report	As Required	As Required

† Minimum Set of Data - Assumes no tool development or special studies.

‡ No DID exits.

§ The Computer Program should be specified as a deliverable.

Note: The same document may be called as an output of different tasks. Updates should be phased accordingly.

provide the necessary information. If this contractual enabling clause is not included, the IV&V effort will be severely hampered because it now depends on the good will of the developer.

3.4.4 Contractual Provisions

Typical contracting approaches are summarized in Table 3-8 for an IV&V effort performed by an ICT. Because of the degree of cost risk

Table 3-8. Generally Accepted Rules for Selecting Contract Types†

Cost-Plus-Fixed-Fee. Appropriate where "level of effort" is required or where high technical and cost uncertainty exists.
Cost-Plus-Award-Fee. Appropriate where conditions for use of a CPFF are presented but where improved performance is also desired and where performance cannot be measured objectively.
Cost-Plus-Incentive-Fee (Cost Incentive Only). Appropriate where a given level of performance is desired and confidence in achieving that performance level is reasonably good but where technical and cost uncertainty is excessive for use of a fixed-price incentive.
Cost-Plus-Incentive-Fee (Multiple Incentives). Appropriate where expectation of achieving an acceptable performance is good but improvement over that level is desired and where technical and cost uncertainties are excessive for use of FPI.
Fixed-Price-Incentive (Cost Incentive Only). Appropriate where confidence in achieving performance is high but cost and technical uncertainty can be reasonably identified.
Fixed-Price-Incentive (Multiple Incentives). Appropriate where improved performance is desired and technical and cost uncertainties reasonably identifiable.
Firm-Fixed-Price. Appropriate where performance has already been demonstrated and technical and cost uncertainty is low.
Firm-Fixed-Price (With Incentives Added). Appropriate where improved performance or schedule is desired and technical and cost uncertainty is low.

† A discussion of basic contract types and provisions will appear in the SAE guidebook entitled Contracting for Software Acquisition. In addition, an excellent discussion of contracting appears in Department of the Army Pamphlet No. 27-153, Procurement Law, January 1976.

and the need for flexibility, a cost type contract with provisions for incentives(either incentive or award fee) seems the most advantageous. This type of contractual relationship allows the government and the IV&V contractor to share the risk.

Typical contractual provisions by performer option for both the IV&V and development contractors are listed in Table 3-9.

Table 3-9. Typical Contractual Clauses †

IV&V Performer	Development Contract Clauses	IV&V Contract Clauses
ICT	• Enabling clause allowing IV&V agency to have access to interim delivery of partially completed computer programs • Proprietary data clause protecting against unauthorized use or disclosures	• Conflict of interest clause excluding firms with organizational conflict of interest from responding to solicitation • Rights in technical data and software clause providing the government with greater rights to software developed under the contract and access to proprietary data and software required to make it operate
ITG	• Rights in technical data and software clause providing the government with greater rights to software developed under the contract and access to proprietary data and software required to make it operate	

† The PCO can assist in development of applicable clauses.

The exact wording of these contractual clauses should be composed by the PCO with technical PO personnel available to act as advisors.

Lately, there has been a tendency to designate IV& V efforts as small business set-asides. The advantages and disadvantages of contracting with a small business are summarized in Table 3-10. The following key factors should be considered during the determination:

- Availability of Experienced Personnel
- Availability of Qualified Tools
- Capability to Respond to Variety of Needs
- Past Performance Experience

Table 3-10. Typical Advantages and Disadvantages of Contracting Small Businesses

Advantages	Disadvantages
• Initial cost could be lower because burden rate is usually lower	• Resource limitations can reduce productivity and elongate schedules
• Corporate commitment is greater because contract represents large part of business base	• The lack of an inventory of proven tools can significantly increase cost and schedule risk
• Responsiveness to PO is often greater because contract is a major business interest	• There is a potential that the firm could fail because of business instability
• More acceptable to prime because he represents less of a threat	• System-level IV&V may require a breadth of specialization not available or affordable to a small firm (especially for quick-reaction technical problem resolution)

3.4.5 Tailored Compliance Levels

Verification and Validation should be tailored to the unique needs of the PO and its overall system requirements. There is no single approach that can be universally applied across the board to structuring the effort. Rather, human judgment is needed to adapt the concepts and

methods to the specifics of the job. This section defines three levels of Verification and Validation that can be used in constructing a responsive IV&V program. Each level is briefly summarized as follows:

- Level 1: Critical Function Identification/Consultant. The IV&V agency directs its efforts toward identifying and monitoring the critical functions. Consultation is provided to work specific problems as they occur and to constructively critique the developer's products. The effort is characterized by review and has limited tool development and testing associated with it.

- Level 2: Design Review/Selected Item Evaluation. The IV&V agency does selected testing in addition to Level 1 activities. The testing is accomplished to independently verify and validate that the critical functions identified during specification verification have been properly mechanized in the code. The IV&V agency may also be tasked to perform additional surveillance in the test area.

- Level 3: System-Level Verification and Validation. The IV&V agency does a complete Verification and Validation of the developer's products. To this end, it performs most of the tasks described in Paragraph 3.4.1.

These three levels of Verification and Validation are compared in Table 3-11.

Selection of the appropriate level is governed by many parameters. Several of the more important of these include:

- Applicable local policy

- Available resources

- Impact of an operational error

- Criticality of application

Table 3-11. Comparison of V&V Levels

Level 1	• Constructively critique developer's documentation • Participate actively in milestone reviews • Identify critical requirements and design problems and recommend solutions • Provide selected technical consultants • Monitor development

Level 2	• Level 1, and
	• Analyze selected critical functions using available tools
	    • Spot check design performance
	    • Evaluate alternate approaches
	    • Conduct limited testing
	• Evaluate critical development test results
	• Perform selected audits
	• Develop selected tools

Level 3	• Levels 1 and 2, and
	• Independently analyze requirements and design
	    • Rederive key algorithms
	    • Confirm technical adequacy
	• Independently test and evaluate operational software
	    • Conduct nominal and stress tests
	    • Identify discrepancies
	• Develop additional tools
	• Provide additional support functions
	    • Special studies
	    • Meetings
	    • Configuration and Data Management

Selection guidelines based upon these four parameters are offered in Table 3-12. The selected level should be applied to:

- All software development phases
- All change requests
- All deliverable products

Table 3-12. IV&V Level Selection Guidelines

Level 1	Level 2	Level 3
• Consider for all applications	• Consider for potentially critical applications	• Consider for all critical applications • Use for nuclear or safety critical applications
• Consider when PO has severe budget limitations	• Consider when PO has staff and budget limitations • Consider when developer has limitations in a specific technical area	• Consider when PO has serious staff limitations • Consider when developer is new to application area and has recognizable limitations
• Consider when cost of error is moderate, but IV&V can be justified	• Consider when error could jeopardize mission success, but cost is hard to quantify	• Consider when impact of error is serious enough to justify cost
• Consider for moderately critical applications	• Consider when there are potential critical requirements that need crystalization	• Consider only when criticality of application is high enough to justify costs

An IV&V program can apply different levels to deliverable products if deemed appropriate.

3.4.6 Cost Estimating

Guidelines for estimating the total cost in terms of percentage of the software development cost for the three levels of Verification and Validation described in Paragraph 3.4.5 are presented in Table 3-13. The guidelines assume that the developer's effort has been appropriately scoped in terms of the following seven factors per deliverable software product:

- Number of instructions
- Programming language

- Type and criticality of software
- Requirements volatility
- Difficulty
- Personnel experience and mix
- Programming practices
- Documentation requirements
- Security level

Table 3-13. IV&V Cost Estimation Guidelines

Level	Cost Relationships	Comments
3	30 to 60 Percent of Development Cost	• Cost Significantly Impacted by: • Hardware Constraints • Software Size and Difficulty • Schedule Inflexibility
2	20 to 40 Percent of Development Cost	• Cost Significantly Impacted by: • Documentation Requirements • Tool Development Needs
1	10 to 30 Percent of Development Cost	• Cost Significantly Impacted by: • Schedule Delays
Cost of all Options is Impacted by: • Developer's Knowledge and Approach • IV&V Agency's Experience and Personnel • Security Level		

Variation in any of these factors during the course of software development can significantly impact the projected cost of the effort. Therefore, it pays to understand the application thoroughly before venturing an estimate.

A typical allocation of resources to the different IV&V activities is shown in Figure 3-2. This distribution will vary depending upon the nature of the work to be performed.

3.5 ORGANIZATIONAL CONSIDERATIONS

This section discusses the managerial function of organization. Organization implies a formalized structure of roles and responsibilities. To arrive at an optimal structure, the VVD must define authority-responsibility relationships and organizationl-contractual interfaces in the VVMP. The VVD must then ensure that the IV&V agency structures its organization properly, and take the steps necessary to reduce interface problems. Each of these subjects will be briefly discussed in the following paragraphs.

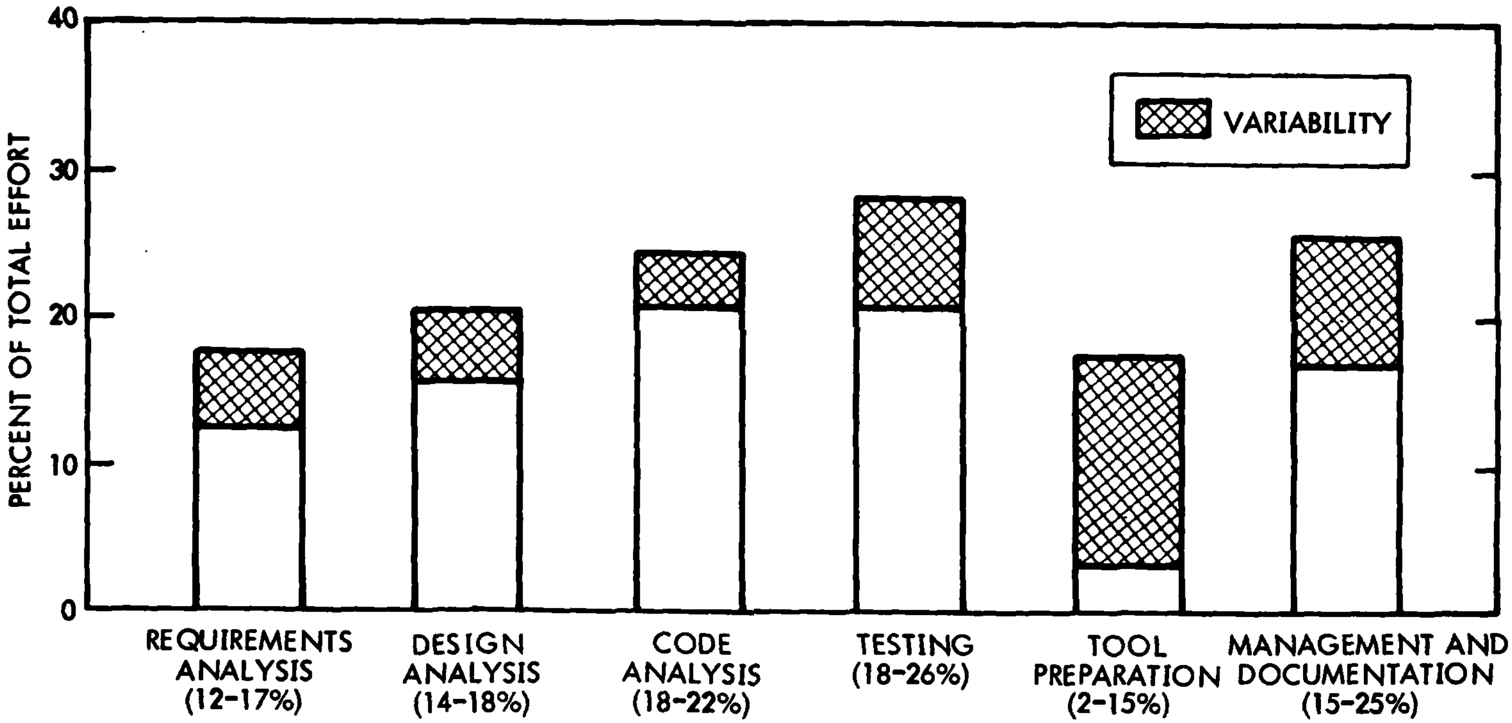
3.5.1 Contractual Relationships

The VVD should decide which contractual relationship he wishes to exist between the IV&V agency, the developer, and the PO for the ICT option. He has the following two basic alternatives available:

- Associate contractor
- Subcontractor

The option should be determined well in advance of the contemplated effort because it dictates which type of procurement will be utilized. The approach selected should be scoped and factored into both the VVMP and the advanced procurement plan.

An associate contractor approach is one where the IV&V and development agencies are directly contracted by the PO to accomplish their respective IV&V tasks. Both contracts contain IV&V provisions and tasks. The PO manages both contracts and is responsible for the interface. The PO management team ideally consists of the VVD and the Software Director who each manage their respective contractors and represent their interests when decisions are made. An integrating contractor arrangement is a special variation of the associate con-



† REPRINTED FROM LOGICON BRIEFING TO SAMSO.

Figure 3-2. Typical IV&V Activity Allocation†

tractor approach. The integrating contractor is responsible for the IV&V of several associate contractor products. He is also responsible for integrating the products into one composite, acceptable system.

A subcontractor relationship is one where the IV&V agency is retained by the developer to accomplish specific IV&V tasks. The IV&V agency typically reports to the developer's program manager in a manner to preserve their independence. The developer manages the IV&V agency and is contractually responsible for their performance. The PO monitors the IV&V agency as it does any other subcontractor. A directed subcontract is a special variation of this arrangement where the PO decides which IV&V house should be hired.

Advantages and disadvantages of these two approaches are similar to those summarized in Section 3.2. The associate contractor approach provides more independence at the expense of additional PO manpower, while the subcontracting approach reduces both independence and PO manpower requirements, but not necessarily the cost.

3.5.2 Interface Control

No matter which contractual option is selected, there will be interface problems between the various agencies involved. People resent the fact that other people are checking their work. Team spirit must be built and "ruffled feathers" smoothed. The VVD must be attuned to the potential for trouble throughout the course of the project and should take immediate remedial action once its need has been identified. The consequences of inaction are dire. Both teams will fight each other instead of working together during the critical front-end phases of the development cycle. The following four approaches have been used effectively to reduce the potential problem:

- Working Groups
- Informal Information Channels
- Third-Party Arbitration
- Management Agreement

Each approach is explained in subsequent paragraphs.

Working groups are formally chartered collections of key personnel who periodically meet to work common problems. The purpose of working groups is to get the working level personnel collaborating with each other in a complimentary manner. The VVD chairs the working group and assigns action items. He records the minutes and controls the interface. Working groups should meet early in the effort and should decide how to handle items that affect common endeavors (e.g., standardizing on a discrepancy reporting system, etc.). As the effort progresses, the group should meet more frequently. Using the group to prescreen and reach agreement on discrepancies has proven a successful means of reducing organizational friction. Joint action items calling for a combined recommended solution to a problem area is a useful approach to team building.

Another approach used to get the players working together is the establishment of informal information channels. All too often the PO acts as the only interface between agencies. This frequently creates an information bottleneck. Allowing for contractor-to-contractor information exchanges through established and controlled channels allows for effective PO supervision and timely contractor action. For example, telecons may be allowed when the PO cannot participate if a problem has to be worked in real time. The telecon must be documented and the record must be provided to the PO.

The third-party arbitration approach should be used only when there is a major technical disagreement between agencies. The PO, development, and IV&V agencies must act as a team. The PO may require the use of a third-party technical expert to act as arbitrator and render the deciding opinion. This has been employed to get a decision made in a manner that doesn't reduce team effectiveness. All parties usually more readily accept the opinion of an outside expert than that of another team member.

The management agreement approach is one where management representatives of the organizations involved mutually agree as to how and when the interfaces will be worked. Agreements are made (both verbal and written) and joint working procedures are published. Typically, these agreements are made at the start of the IV&V effort. This helps establish what the roles and responsibilities of the team members will be.

3.5.3 Organization Structure

A typical functional organizationl structure for an IV&V agency is illustrated in Figure 3-3. The function of each of the four groups and how it changes as the project transitions from development to operations and maintenance is explained in the following text.

The V&V Tools Group is responsible for the following functions:

- Tool Development
- Tool Modification
- Training
- Tool Maintenance

Because tools should be user-oriented, each of the other three groups provide requirements to the Tools Group and participate in acceptance testing and training. The Tools Group is used throughout the life cycle to provide a level of support.

The V&V Analysis Group is responsible for the following functions:

- Specification Evaluation
- Test Review
- Special Studies

Again, they are active throughout the life cycle. Some of the key personnel from this group become the nucleus of the Performance Evaluation Group. In this way, knowledge gained can be used/applied during code evaluation efforts. As transition to operations and maintenance nears, they become actively involved in the analysis of change

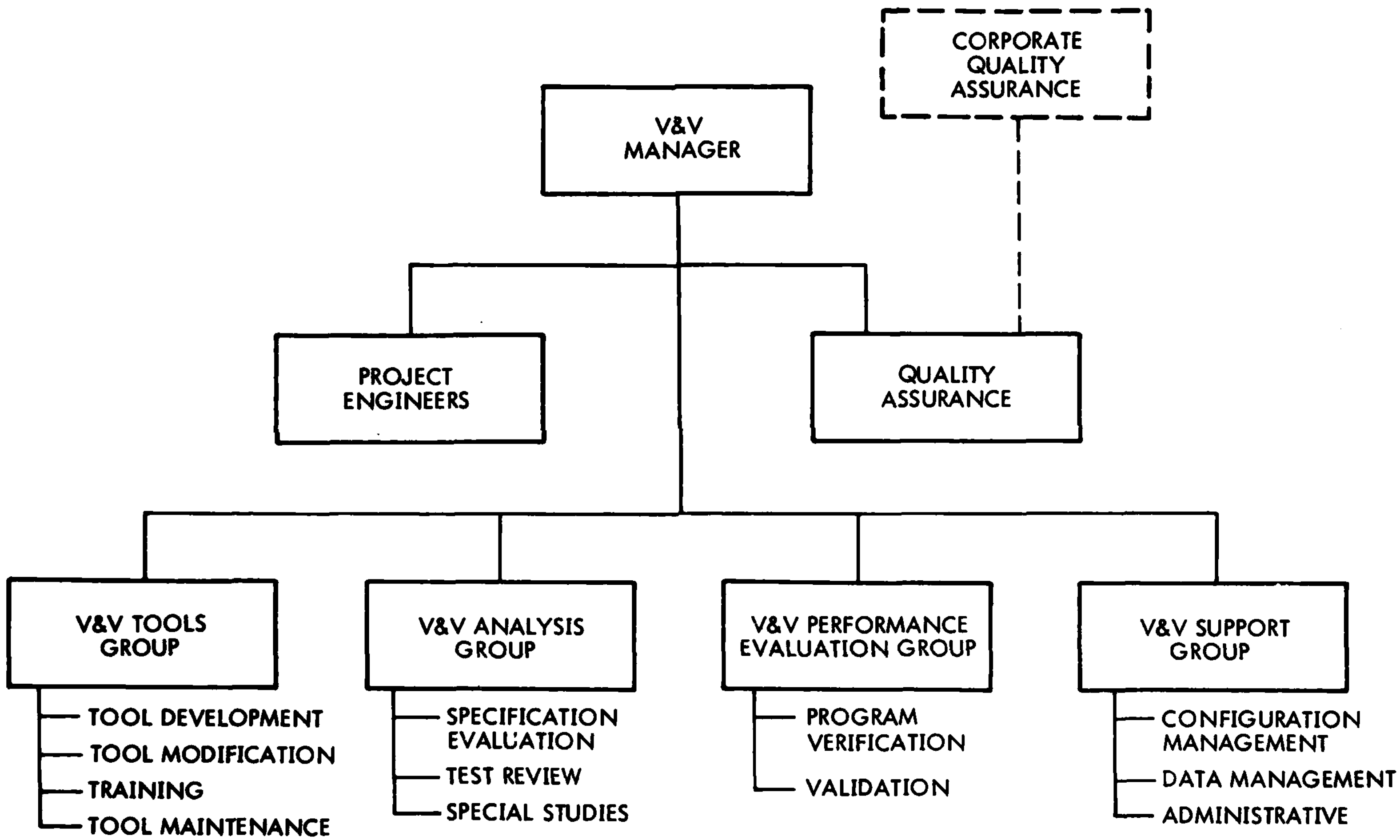


Figure 3-3. Typical IV&V Organizational Structure

requests. They provide the PO with an independent assessment of the impact of the change and evaluate whether or not IV&V is warranted.

The V&V Performance Evaluation Group is responsible for accomplishing program verification and software validation. They work with the developer and independently test and evaluate the code. During operations and maintenance, they reverify and revalidate the program as changes are introduced and as circumstances require.

The V&V Support Group is responsible for providing configuration management, data management, and administrative support as required. Administrative support includes financial management functions of the IV&V contract.

Because quality assurance like IV&V requires an independent reporting channel, it is broken out as a separate organizational entity reporting to the V&V Manager. The quality assurance manager should also report on a dotted line to some higher level quality assurance authority so that leverage can be exerted on the V&V Manager.

In the case where there are multiple packages being verified and validated, a project engineer should be assigned the responsibility of ensuring that tasks are completed on time and within budget. Project engineers should report to the V&V Manager. Because there are different discipline mix requirements during the project lifetime, the nature of this staff will change over time.

3.6 PERSONNEL CONSIDERATIONS

Skilled personnel are key determinants of success in an IV&V project. The people employed must be experienced problem solvers, designers, implementers, testers, managers, expediters, and communicators. They must be well grounded in theory and practice and should be familiar with management and economic issues. They should fully understand the job, the PO, and be capable of working in a pressure-filled environment. They must be tactful and politically astute. They must possess a wide range of skills and be able to employ them under difficult circumstances.

The skills required to successfully verify and validate software vary depending on the nature of the job. An aircraft avionics project skills matrix displaying typical skills by task is illustrated in Table 3-14. It is useful to construct such a matrix and utilize it to determine:

- What skills the project needs.
- What the staff (PO and IV&V agency) strengths and weaknesses are.
- What the personnel training needs are.
- Whether selective recruiting is warranted.
- Whether consultants or subcontractors can be effectively employed to overcome staff weaknesses.

If a competition is held for the IV&V job, the PO can have the contenders fill out a skills matrix. The PO can then evaluate the completed matrix during the selection process.

The skills matrix should be updated during the project because the mix changes during the life cycle. Early in the life cycle, systems oriented skills predominate. Later on, software test and evaluation skills become the most important. Personnel planning is necessary to cope with the problem of getting the right people to do the work. It is the responsibility of the PO to ensure this planning is accomplished in a timely manner.

3.7 MANAGEMENT CONTROL

This section discusses the managerial function of control. Control assists the VVD in making things happen as planned. The basic control process involves establishing standards, measuring performance against these standards, and correcting deviations from standards and plans. Standards for management control which should be provided in the VVMP include:

- Work Authorization
- Budgeting
- Performance Measurement
- Variance Analysis (Schedule and Cost)
- Accounting

Table 3-14. Skills Matrix for a Typical Aircraft Avionics Project

Tasks	Skills												
	Application-Oriented					Methodology-Oriented		IV&V-Oriented			Management-Oriented		
	Real-Time Software Develop.	Equation Eval. and Validation	Controls and Displays	Computer Architecture	Total System Under-standing	Modern Program. Techniques	Config. Mgmt. Practices	Tool Costs and Benefits	Analysis Techniques	Test and Eval. Approaches	Team Building	Life Cycle Cost Analysis	Planning and Control
System Specification Verification		X	X	X	X				X		X	X	X
Tool Development and Maintenance	X			X	X	X	X	X		X	X	X	X
Software Requirements Verification	X	X	X	X	X	X		X	X		X	X	X
Software Design Verification	X	X		X	X	X		X	X		X	X	X
Program Verification	X			X	X	X	X	X	X	X	X	X	X
Software Validation	X		X	X	X	X	X	X	X	X	X	X	X

Key tools used to enforce these standards and to measure performance against them include reports, reviews, control room, and libraries. Each of these tools are briefly discussed in the following paragraphs.

3.7.1 Reports

Information is the management link between the means and the ends. It is studied, analyzed, organized, stored, summarized, and/or displayed. Information is different from data. Data is raw information often described as "facts in isolation". Information is the aggregate of data or facts organized into knowledge in an intelligent fashion. Management thirsts for information and drowns in data!

Valid and timely information describing the technical and programmatic (cost and schedule) progress of the IV&V effort is needed for proper management control and action. Data must be collected and information reported in a fashion that makes the VVD's job easier. Suitable provisions need to be incorporated to provide the details needed for decision making.

Reports convey information to the VVD. The following five types of information should be considered when arriving at a report structure for the IV&V effort:

- General project summary
- Status and progress summary
- Resource utilization
- Error history
- Deficiency status

General project summary information includes high level data which describes the project, its purpose, and the IV&V approach. It is used to provide new personnel and the uninformed with the "big picture" summary of the whats, whys, and how. It is often useful to have the IV&V agency prepare a brochure to report this information. Periodic updates are recommended to keep the brochure current.

Status and progress summary information includes disclosure of technical, schedule, and financial status periodically by task. Significant accomplishments, milestones, and product deliveries are all noted. A listing of open action items and problem areas is normally included. Exception reporting where exceptions are flagged to indicate to the VVD where action may be required is useful.

Resource utilization and error history information includes statistics that reveal trends and isolate potential problem areas. Timing and sizing projections can be used to identify growth trends. Scheduling difficulties can be identified by analyzing computer utilization statistics. Error statistics which identify code segments that have had a history of problems and are error prone are useful in determining where to concentrate IV&V activities.

Deficiency status information identifies deficiencies by program package and the actions taken to correct them. It helps flag past-due problem areas. It is an important IV&V tool because it ensure deficiencies are acted on and closed.

Requiring the IV&V agency to deliver a monthly progress report with separate sections dealing with each of the five types of information noted above has proven to be a useful management tool. The advantages of this approach are that it reduces the number of deliverables and provides the data in a timely manner.

Standardizing on one discrepancy reporting system has proven to be another useful concept. A standardized discrepancy report and reporting procedure reduce the confusion associated with having several agencies review and comment on multiple products. It also reduces cost because only one agency has to be tasked by the PO with the job of being the custodian of discrepancy reports.

3.7.2 IV&V Contract Reviews

The second major management control tool is reviews. The IV&V agency should conduct both technical and management reviews to assess progress and work specific actions. Because guidelines for the establishment and conduct of technical reviews has been published in the SAEG on Reviews and Audits, this paragraph will treat managerial reviews only.

Project reviews are conducted on a regular basis at all levels of management. Specific reviews that should be considered by the VVD include:

- Informal periodic reviews with the IV&V Project Manager and key personnel.
- Formal monthly review on project status.
- Formal monthly review of subcontractor performance, if appropriate.
- Quarterly reviews with upper management.

These reviews should be directed toward ensuring that all levels of PO and contractor management are advised on the progress and status of the project. They provide the VVD with the opportunity to identify and discuss potential risk issues and how to avoid or minimize them.

3.7.3 Project Control Center

The project control center represents another tool for managerial control. A control center is a room physically established at the IV&V agency location. In the room, the following items are displayed:

- The current status of relevant IV&V and prime tasks including task descriptions, organization, schedules, methodology, required inputs and outputs, current activities, and staffing.
- The critical path in the form of an activity network.
- The current status of risk areas and action items.
- The summary status of all deficiency reports.

The control center is used for review meetings, technical interchange, and technical direction meetings. The control center will always remain open for VVD inspection so that the current status of the project will remain completely visible.

3.7.4 Libraries

The final management control tool to be discussed is libraries. Libraries are depositories established to store programs and data. They provide powerful support to both the program development and IV&V process in the following areas:

- Provide up-to-date descriptions of the programming system's status via up-to-date descriptions of its programs, test data, discrepancies, and/or documentation.
- Collect and report relevant management information.
- Control the integrity and security of the data stored in the depository.
- Support the status accounting and configuration control aspects of configuration management.
- Allow multiple parties (each with a need to know) access rights to the current programming system (public instead of private).

Both the development and IV&V agencies should establish their own library system. This makes each accountable for its products. Both agencies should then make provisions to allow the other and the PO to selectively access the library as the need occurs. These support an open interchange of information among all parties involved. As discussed in Section 3.5, such an open interchange is essential in order to get all agencies working effectively as a team. A team attitude will make everyone more productive and will reduce the amount of organizational conflict involved.

3.8 TOOLS AND TECHNIQUES

This section discusses the following issues relative to IV&V tools and techniques:

- Tool Selection Methodology (Paragraph 3.8.1)
- Tool Requirements (Paragraph 3.8.2)
- Tool Development, Qualification and Maintenance (Paragraph 3.8.3)
- Tool Ownership and Delivery (Paragraph 3.8.4)

The discussion that follows is structured to assist the VVD in selecting, developing, and fielding a set of tools and techniques that fulfill the life cycle needs of his program.

A glossary of available tools and techniques is provided in Appendix B.

3.8.1 Tool Selection Methodology

Verification and Validation tools can automate tedious, costly, error-prone processes. They can also be used to economically capture technical data that may not be available using manual techniques. These benefits do not come without their attendant costs. Potential costs and benefits should be assessed prior to making a decision relative to which tools and techniques should be utilized.

A tool selection methodology is illustrated in Figure 3-4. The approach taken is to conduct a needs assessment to identify potential problems associated with the application and candidate tools and techniques that can effectively deal with them. Once the needs have been quantified, desired capabilities can be determined and candidates evaluated in terms of Life Cycle Cost (LCC). In order to determine if there is a genuine need for a specific tool or technique, costs and benefits must be quantified and a projected Return cn Investment (ROI) should be calculated. Typical costs that should be determined include:

- Tool development and/or modification costs
- Tool installation and demonstration costs
- Tool operation and maintenance costs
- Training and consultation costs

Typical benefits attributable to tool use include:

- Reduced manpower
- Improved diagnostic capability
- Quicker turnaround
- Increased efficiency

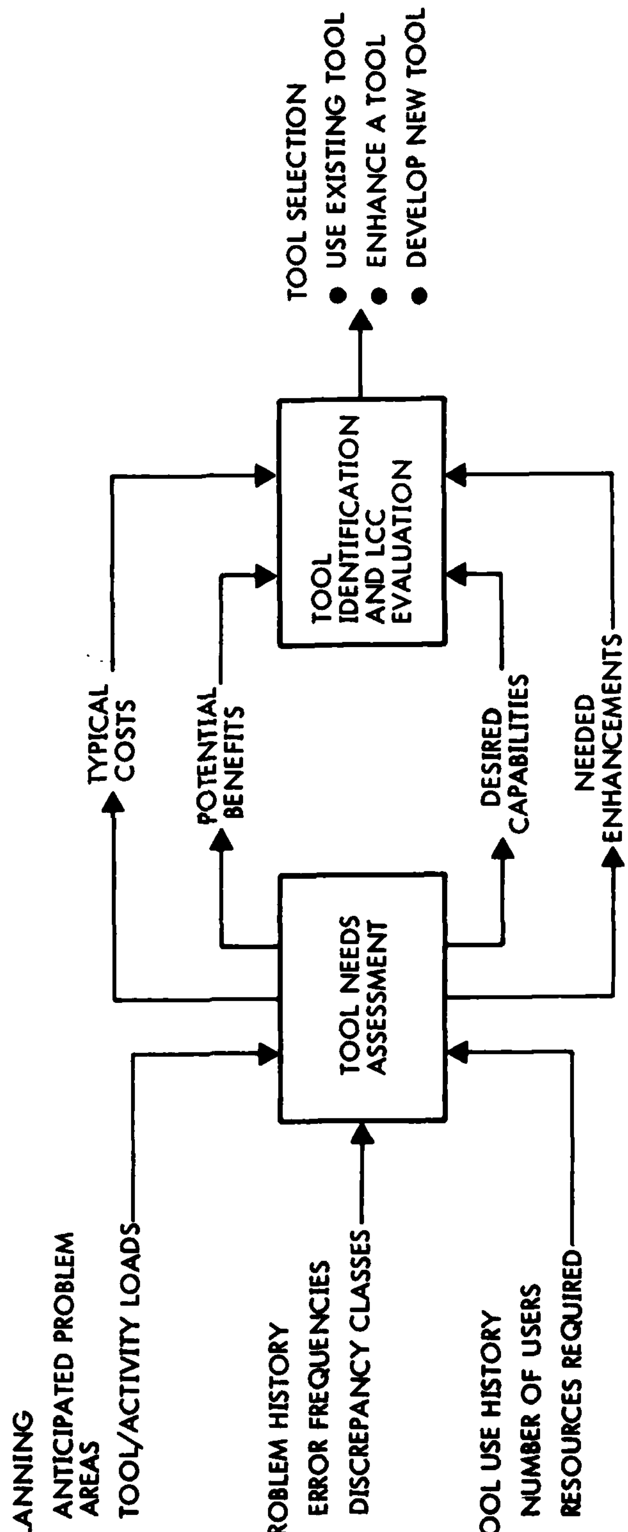


Figure 3-4. Tool Selection Methodology

Costs and benefits for each tool should be quantified in terms of dollars and cents. Then, a projected ROI for each tool should be determined. Finally, the individual ROI's should be compared against the minimum attractive ROI for the project in order to determine which of the candidates is acceptable. Experience indicates that an ROI of 15% seems admissable to most projects.

For competitive procurements, the PO could require those responding to submit a tools study. This approach has the advantage of allowing the VVD to evaluate the offeror's tools as part of the source selection process using the requirements in the next paragraph as a guide.

3.8.2 Tool Requirements

Verification and Validation tool requirements impact selection. Typical requirements that will be discussed in this paragraph include:

- Tool independence
- Tool schedules
- Tool documentation
- Tool configuration management
- Tool portability and reuse

The IV&V agency should not use the developer's tools. They should develop a separate set that preserves the independence of their analysis. For example, errors are often introduced into the operational code by the developer's tools. If the IV&V agency used the same tools, these errors would never be caught. The IV&V agency should also strive to make their tools complement, not duplicate, the developer's tools. For example, the IV&V agency could use a software test bed to test the program the developer is checking out using a hardware test bed. This approach stresses the program in multiple dimensions.

The IV&V agency's tool selection is normally dicated by the developer's schedule. Tool schedules must be phased to ensure that the right tool is available when needed. Because analysis tools are

needed early in the effort, their development represents a potential risk that should be evaluated by the VVD. Large simulators represent another risk area. These programs are large and costly to develop. The IV&V agency should use existing simulators whenever possible to reduce the risk.

Tool documentation can significantly increase the cost of an IV&V effort. The VVD should determine his life cycle needs before arriving at his data requirements. Full military standard documentation is not always required. The documentation requirements noted in Paragraph 3.4.3 are the minimum considered acceptable.

Tools should be baselined and placed under configuration control and maintained as a project product. Again, strict configuration management under the auspices of the PO is expensive and should not be employed if not needed. The IV&V agency should place its tools under internal configuration control regardless of the degree of formal procedures specified by the PO. This provides for product integrity at minimum cost to the government.

Existing tools should be reused whenever feasible to reduce the cost and the development risk. The IV&V agency with PO approval should determine whether existing tools can fulfill the needs identified during the tool selection process. The IV&V agency should collect experience data about candidate tools and should quantify LCC before recommending a make/buy decision. Tool efficiency is a major consideration. Many existing tools are expensive to use repeatedly for long runs. Such tools should either be enhanced or replaced by new ones. Support tool requirements must be kept in the proper perspective. Tools are not the end product of the IV&V effort.

3.8.3 Tool Development, Qualification and Maintenance

The IV&V agency should treat their tools as software products and should develop them using a disciplined methodology. The PO should address the subject of tool development methodology in the VVMP. The

IV&V agency should devise a tool development plan based on his assessment of the user's need, the VVMP's provisions, the developer's schedule, and life cycle cost analysis. Major methodology items that the IV&V agency should consider for incorporation into its plan include:

- Use of a sequential life cycle process where necessary documents and reviews are held at important points in the cycle.
- Use of modern programming techniques (e.g., structured programming, peer reviews, top-down development, etc.)
- Use of approved high order languages
- Establishment of a set of standards and procedures for the effort.

The plan should address both new developments and modifications. The plan should be put into action and software (tools) should be developed in an orderly manner.

After each support tool program has been developed, it should be subjected to qualification testing. Formal procedures for incorporating changes (e.g., repairs, enhancements, etc.) should be enforced and the software should be kept current.

The following three step acceptance procedure has been successfully used on a number of IV&V projects:

1. The IV&V agency developing the tools demonstrates them according to an approved acceptance test plan.
2. The IV&V agency trains the user and helps him become familiar with operating the tools.
3. The user demonstrates the tools to the IV&V agency using the approved acceptance test plan. Results are compared with previous tests and if they correlate, the system is released for use.

Other procedures can be employed to achieve the desired results.

3.8.4 Tool Ownership and Delivery

The VVD must address the emotional issue of tool ownership in his VVMP. Ownership infers that the government owns the greater rights to the tools and the software required to make them usable. Many firms don't want to give the government rights to their tools because:

- The tools give them a competitive advantage.
- The tools represent a capital expenditure for which government remuneration is not adequate.
- The tools are poorly documented and hard to use.
- The tools are not portable.

The government wants rights to the tools because:

- There is an identified need in the O&M Phase for government usage of the tool.
- Cost leverage is increased for future procurements. Because tools provide a competitive advantage, providing them as Government Furnished Equipment (GFE) in procurement packages levels the competition. Unfortunately, it also increases the risk because the government is now liable for the tool's performance.
- There is a potential need in the future for government acquisition of the tool. For example, the government may want a third party to perform independent analysis in some critical area where neither the developer nor the IV&V agency is qualified (e.g., survivability, nuclear effects, etc.). The government may want to GFE tools to reduce the cost of this analysis to an affordable level.

When there is an identified need for a tool, it should be delivered with full rights to the government. The IV&V agency should install the tool on the host facility and demonstrate that it fulfills the agreed-upon acceptance criteria. Fair and equitable return should be negotiated between the government and the IV&V agency for capital expenditures. Pre-determination agreements which identify the government's greater rights to each modified, used, or deliverable software package should be negotiated and agreed-to prior to contract award.

When there is a potential need for a tool, the PO should consider negotiating an agreement which in essence states that the government has the option to purchase greater rights, delivery, installation, and training for some agreed-to price throughout the contract period. Clauses of this type allow the PO to purchase tools at a later date when the need becomes real.

The PO should always negotiate greater rights. The PO can induce a contractor to provide the government with unlimited greater rights by providing him with the rights to license the tools commercially. Such an approach is financially atrractive to both parties.

There are several pitfalls associated with tool ownership. First, the government assumes full liability for a tool's performance once delivered and accepted. If the tool is going to be provided as GFE to some third party, maintenance agreements will have to be negotiated. Second, most tools have limited portability. The costs of rehosting the tool on another machine should be considered when the cost/benefit analysis is conducted. Third, learning to use the full capabilities of a new tool is often a long and arduous task. Training and consultation services should be considered to help the user tap the full capabilities available.

4. SPECIFIC GUIDANCE FOR THE CONDUCT OF VERIFICATION, VALIDATION AND CERTIFICATION

General guidance in planning, organizing, staffing, directing and controlling a Verification and Validation effort was provided in Section 3. This section provides the Air Force Program Office (PO) and its engineering personnel with more detailed guidance into both the technical and management aspects of the job. Descriptions of the tools and techniques referenced in this section can be found in Appendix B.

4.1 SYSTEM SPECIFICATION VERIFICATION

System specification verification is the V&V activity conducted to ensure that the system/system segment being considered will meet its mission goals and objectives. Once this activity is completed, the subsystem requirements can be developed in a logical manner with assurance that there is a clear and accurate description of the systems concept.

System specification verification occurs during the Validation Phase. It takes the system specification and/or data system specification and determines whether the stated requirements are a clear and accurate translation of the user's need as stated in the SON document. The tasks typically performed by the IV&V agency during specification verification are listed in Table 4-1. The results of these activities culminate at the System Design Review (SDR).

4.1.1 Responsibilities - Government and Contractors

The Validation Phase begins with a preliminary System Specification, Test and Evaluation Master Plan (TEMP) and Verification and Validation Master Plan (VVMP) which are normally developed by the Program Office. The development contractor's first task is to update the System Specification and TEMP so that they are compatible with the approved system engineering concepts and to prepare the System Engineering Management Plan (SEMP). The contractor then begins the

Table 4-1. System Specification Verification Tasks

- Prepare Contract or V&V Master Plan
  - Identify Tool and Test Requirements
- Identify/Evaluate System Requirements Allocation to Software†
  - Performance Budgets
  - Interfaces
  - Operational
- Evaluate/Determine Preliminary Computer Resource Requirements†
  - Operational or Target Computer
  - Test and Evaluation Support Systems
  - O&M Support Systems
- Evaluate/Determine Support Tool Requirements
- Identify/Specify IV&V Interface Requirements with Developer
  - Intermediate and Final Products
  - Discrepancy Change Control System
- Participate in User/System Design Reviews (SDR)
  - Assist and Understand His Needs
- Identify/Evaluate Operational Software Requirements†
  - HW/SW Interfaces
  - Performance Budgets
  - Man/Machine Interfaces
  - Functional Processing
  - Data Base - External Interfaces

    } • Testable • Complete • Consistent • Feasible • Traceable
- Review/Evaluate Developer's Documentation
  - Computer Program Development Plan
  - Draft Part I Development Specifications

† See Requirements Analysis and Specification Guidebook for greater detail.

task of refining the system concept and allocating requirements to subsystems and then to CI's and CPCI's. The process continues with the contractor conducting trade studies which help reduce the risk of the system design. The development contractor holds a SRR and SDR during this timeframe so that the government (PO and users) can review and gain visibility into his progress. The main products developed during this phase are an authenticated System Specification, TEMP, plans, trade studies, preliminary specifications and Interface Control Documents (ICD's). A detailed discussion on the reviews can be found in the SAEG's on Quality Assurance and Reviews and Audits.

The IV&V agency is typically brought on contract just after the PO approves the final SRR minutes. Their first major task is to prepare a Contractor Verification and Validation Master Plan (CVVMP) which reflects the provisions of the PO's VVMP. The IV&V agency then starts a detailed review of the development contractor's products and reports their findings to the PO. The IV&V agency initiates their tool development activity and their test planning during this period. Their participation in the SDR culminates their independent confirmation of the feasibility of the requirements.

The PO monitors progress and reviews and approves products produced by both participants. The PO attends reviews, approves minutes, and assigns action items. The PO works with both the development contractor and the IV&V agency to provide task direction, establish team spirit and proper working relationships. They review deliverables and evaluate their technical adequacy and acceptability.

4.1.2 Concepts and Methods

Specification verification is concerned with analyzing and evaluating the system specification requirements and their allocations in detail. Detailed requirements analyses are conducted using analytical modeling, simulation, and prototyping (see Appendix B for explanation) to critically evaluate the proposed conceptual approaches to system mechanization. Preliminary subsystem relationships are reviewed to ensure satisfaction of appropriate performance, functional, and operational requirements.

Requirements are segmented in sufficient detail to determine whether the identified design approaches can realize them with acceptable risk.

The IV&V agency should direct their efforts toward evaluating the following three areas during specification verification.

- Risk
- Technical Feasibility
- Supportability

Trade studies are conducted to evaluate alternative system concepts in terms of cost/risk. Typically, the attitude "let the computer do it" prevails. As a result, the cost/risks are not fully evaluated. The IV&V contractor must appraise the PO of the consequences of trades. They must quantify risk in terms of a range of direct (dollars) and indirect costs (schedule). For example, most airborne systems have equation trade studies investigating different guidance or navigation schemes. These trades are the precursor to the derivation of the equations that go in the Part I Development Specification. Because the equations are the backbone of performance, acceptable engineering solutions (accuracy, speed, etc.) must be verified for a variety of nominal and off-nominal situations. A major change in philosophy could impact hardware selection and software cost. Coding the equations in FORTRAN and executing them with models of the environment using an engineering simulation has proven to be a successful method of proving feasibility early. Other risk reduction techniques include simulation and prototyping. If you've never done it before, it normally pays to build a "quick and dirty" prototype to prove the concept.

Technical feasibility of the functional allocations to hardware, software, firmware, and operator procedures (could be implemented by the pilot) is the next item to be evaluated. The typical philosophy is "let the software do it if it is tricky." With the advent of cheap hardware and firmware, this is not always the right way to go. The IV&V contractor should evaluate the feasibility of the allocations in terms of life cycle costs and appraise the PO of his findings. Analytical modeling

can be used to investigate the complexities of real time systems. System simulations which functionally model the architecture can be employed to do hardware/software tradeoffs to assist in allocation. Performance evaluation and workload measurement aids have been used effectively in evaluating performance of existing hardware and software in architectural evaluations.

Another key problem is the tendency of the developer to concentrate on the operational software. Typically, little attention is given to support software used in the development facility, support and test equipment. The availability of critical checkout equipment or a compiler can drive the schedule. The IV&V contractor should ensure that the developer's Computer Program Development Plan adequately addresses these issues. The IV&V agency should spend as much time as necessary (depending on criticality) to ensure that the PO is appraised of the risks and possible consequences in these areas.

4.1.3 Products and Problems

The typical products and potential problems associated with the specification verification activity are illustrated in Table 4-2.

4.2 SOFTWARE REQUIREMENTS VERIFICATION

Requirements verification is the V&V activity conducted to ensure that the software requirements can accomplish their allocated system requirements. Its primary aim is to identify ambiguous, ill-defined and technically inadequate software performance and design requirements as early in the process as possible.

Requirements verification occurs during the Validation Phase. It ensures that the computer program development specifications adequately reflect the computer-applicable portion of the system specification. The tasks typically performed by the IV&V agency during this period are listed in Table 4-3. The major software product of this activity is a set of authenticated specifications which become the allocated baseline for the Full Scale Development Phase.

Table 4-2. System Specification Verification Products and Problems

Products	Problems
Developer	
• Computer Program Development Plan (CPDP)	• Creating a Team
• System Engineering Management Plan (SEMP)	• Developer mistrust of IV&V agency
• Contractor Test And Evaluation Master Plan (CTEMP)	• PO must manage delicate interface
• Configuration Management Plan (CMP)	• Lack of Information Exchange
• System Specification	• PO must handle formal exchange
• Trade Studies	• Informal exchange necessary
• Draft ICD's	• Software Requirements at the Mercy of Other Disciplines
• Draft Part I Development Specifications	• Software personnel must be involved early (both developer and IV&V)
IV&V	• Interdisciplinary working groups solve common problems
• Contractor V&V Master Plan (CVVMP)	• Failure to Evaluate Full Consequences of Trades
• CPDP (Tools)	
• CMP	
• Trade Studies	
• Part I Development Specifications (Tools)	

Table 4-3. Software Requirements Verification Tasks

- Update/Evaluate Computer Resource Requirements
- Identify Traceability of Requirements to System Level Documents
  - Consistent
  - Complete
  - Adequate
  - Testable
- Evaluate Functional Performance
  - Accurate
  - Efficient
- Review/Evaluate Developer's Documentation
  - Part I Development Specifications
  - Draft Standards and Procedures Manual
  - Updated Computer Program Development Plan
- Prepare Preliminary IV&V Test Plan
- Develop/Modify V&V Tools
  - Conduct Required Reviews
- Participate in Software Requirements Review

4.2.1 Responsibilities - Government and Contractor

The development contractor's responsibility during the period from SDR to the end of the Validation Phase is to (1) revise his Part I Development Specifications and ICD's based upon approved SDR actions and continuing requirements definition activities and (2) support the conduct of a Software Requirements Review. The requirements specified should be finalized when they are sufficient to form an allocated baseline for design. The requirements should then be reviewed at a contractor conducted Software Requirements Review where an action plan for approval of the Part I Development Specifications and subsystem ICD's should be formulated. The approved Part I Development Specifications and subsystem ICD's will form a part of the Full Scale Development RFP package and will scope part of the ensuing contractual effort.

The IV&V agency's responsibility during this period is to evaluate the developer's products to ensure their technical viability with regard to the computer-applicable requirements of the system specification. Requirements are analyzed and are sometimes independently derived in order to verify the developer's allocations which form the basis of design. The IV&V agency is as responsible for the requirements as the developer. They must do everything necessary to give the PO their assurance that the Part I Development Specifications and other supporting documents are technically sound.

The PO continues to monitor progress and review and approve products produced by both participants. The PO attends reviews, chairs working group meetings, institutes technical interchange meetings, approves minutes and assigns action items. The PO's major responsibility is to make sure that the requirements get defined and specified in a form appropriate for baselining. The PO must also make sure that the schedules are maintained for support and checkout equipment needed for software production. If the requirements in the Part I Development Specification are ill-defined, the PO should extend the definition activity. In making their decision, the PO must listen to both the developer and the IV&V agency. Baselining too soon can lead to large cost overruns. Baselining too late could cause delays and other problems.

4.2.2 Concepts and Methods

Requirements verification is concerned with evaluating the developer's Validation Phase products in detail in order to confirm that they form an appropriate baseline for the Full Scale Development Phase. The Part I Development Specification are evaluated to ensure their requirements are consistent, complete, testable, and technically adequate. The evaluation is directed towards answering the questions posed in Table 4-4 and in the SAEG on Requirements Analysis and Specifications. Evaluation tools and techniques employed to answer these questions are described in Appendix B and include analytical modeling, simulation, algorithm evaluation testing, requirements languages, tracers, and prototypes.

Table 4-4. Software Requirements Verification Checklist

- Are all functional, interface, and test requirements completely specified in quantitative terms?
- Are there any potential problem areas in fulfilling the requirements?
- Are the requirements logical, consistent, testable, traceable, and understandable?
- Are the requirements sufficient to realize both the system and subsystem objectives?
- Are all input, output, and processing requirements identified and specified for each function without ambiguity?
- Are all hardware and software interfaces identified?
- Are the data base and data requirements clearly stated?
- Are acceptance criteria specified for each requirement?
- Have the equations been scientifically verified?
- Have the human engineering aspects been addressed adequately?
- Is there early and continued emphasis on test planning?
- Are the objectives and stages of testing described?
- Do timing and sizing estimates have sufficient margin?

The IV&V agency should direct their efforts towards evaluating the following four areas during requirements verification:

- Technical Adequacy
- Criticality
- Testability and Supportability
- Timing and Sizing

The primary objective of requirements verification is to confirm the technical adequacy of the requirements. The specifications are first evaluated for completeness, consistency, and traceability to the system specification. Special requirements language systems have been developed to effectively automate part of this process. Then, the detailed functional and performance requirements are analyzed in great detail. Some IV&V approaches that have proved successful in the past include:

- Use of scientific simulations enhanced with more sophisticated models (i.e., sensors, vehicle, atmospheric, etc.) to verify the accuracy of the equations in their engineering form for realistic environments.
- Use of functional simulations to evaluate interrelationships between functions and functional performance (i.e., timing, sequencing, etc.)
- Use of protypes to validate requirements derived for functions for which little or no history exists (e.g., a new redundancy management technique).
- Use of capability matrices or N^2 charts to trace functions or their interfaces vs other requirements.

An essential part of functional analysis is determination of critically. In many instances, the cost of IV&V prohibits its cost effective application to the entire program. Only those functions deemed critical, then, are subjected to an IV&V. Candidates for IV&V could include such functions as a terminal guidance function for a homing interceptor, a range safety destruct function, a critical bombing algorithm, a precision radar tracking function on an aircraft, and an entire flight program for a launch vehicle.

Another area of concern revolves around the tendency of the developer to concentrate on performance at the expense of testability and supportability. The IV&V agency should ensure, using analyses, that every requirement stated in the specification is testable. This requires a detailed examination of the TEMP and test requirements section of the Part I Development Specification. Having the IV&V agency review test documentation is controversial. One school says they shouldn't because it will bias the IV&V test approach. Another school says they should because only then can the IV&V program be made complementary to the developer's approach. In addition, the specifications should be evaluated to ensure they are consistent and compatible with the provisions of the PO produced Computer Resources Integrated Support Plan (CRISP). Supportability is as important a consideration as testability.

The final area of concentration for the IV&V agency is that of timing and sizing. The IV&V should independently derive timing and sizing estimates based upon their experience. These estimates can then be compared with the developer's and disparities should be examined before budgets are established.

4.2.3 Products and Problems

The typical products and potential problems associated with the requirements verification activity are illustrated in Table 4-5. The questionable adequacy problem noted in Table 4-5 concerns the issue of baselining inadequate requirements. It is often better to delay approval until the Full Scale Engineering Development Phase rather than prematurely authenticating the Part I Development Specifications.

4.3 SOFTWARE DESIGN VERIFICATION

Design verification is the V&V activity conducted to ensure that the software design represents a clear, consistent, and accurate translation of the software requirements. Its primary aim is to confirm the fact that the recommended design will do the job specified in the Part I Development Specification. It does not attempt to redesign. Instead, it seeks to identify inadequacies in the design and test approach before implementation starts.

Table 4-5. Software Requirements Verification Products and Problems

Products	Problems

<u>Developer</u>
• Part I Development Specifications
• Subsystem ICD's
• Standards and Procedures Manual (Draft)
<u>Program Office</u>
• Draft Computer Resources Integrated Support Plan (CRISP)
<u>IV&V</u>
• Updated Tool Documentation (Existing Tools)
• Draft Part II Product Specifications (Tools)
• IV&V Test Plan
• Reports
    • Timing and Sizing
• Discrepancy Reports (DR's)
	• Lack of Team Work
	    • Continual PO pressure to maintain interface and exchange information
	• Questionable Adequacy of Software Requirements
	    • Pressure to create baseline before full scale development
	    • Other disciplines slow in deriving their requirements
	    • Critical decisions that impact software delayed (e.g., computer selection)
	• Multiple Discrepancy Reporting Systems
	    • Standardize on one early
	• No Preliminary Test Approach Developed
	    • Must require early full scale development delivery of test documentation

Design verification is a Full Scale Engineering Development Phase activity. It takes the Part II Product Specification in two versions (preliminary and draft) and ensures that the evolving design adequately satisfies the provisions of the Part I Development Specification. The tasks performed by the IV&V agency during this period are listed in Table 4-6. The major software product of this activity is a set of draft Part II Product Specifications which form the basis of coding.

4.3.1 Responsibilities - Government and Contractors

The development contractor's responsibilities during the period starting with the beginning of the Full Scale Engineering Development Phase and ending at the CDR is to (1) formulate a software design and test concept, (2) develop a detailed design using this concept that fulfills the requirement of the Part I Development Specification and, (3) support the conduct of a Preliminary Design Review (PDR) and a Critical Design Review (CDR). The Full Scale Engineering Development Phase should start with authenticated Part I Development Specifications and ICD's. These should be updated and an acceptable design and test approach should be developed that meets their intent. The PDR should provide an action plan for approving the approach which establishes the design architecture for each CPCI. This architecture is then refined successively until it is of sufficient detail to commence coding. A CDR is then held to provide an action plan to finalize the design and test procedures. The CDR data package typically consists of an agenda, draft Part II Product Specifications, draft test procedures, draft users manual, and draft version description documents.

The IV&V agency's responsibility during this period is to evaluate the developer's products to ensure their technical viability and to contribute to the design refinement process. The design is checked for logical consistency and completeness. Key algorithms may be either simulated or rederived in order to assess their technical adequacy. The IV&V agency must do as much analysis as is necessary to independently verify the design implementation. They provide the PO with their assurance that the design is technically sound and that its critical components will do the job.

Table 4-6. Software Design Verification Tasks

- Identify/Evaluate Traceability of Design to Part I Specification
- Analyze Design Structure
  - Executive Structure
  - Logic/Control Flow
  - Data Base - Internal and External
  - Error Recovery
  - Modularity
- Evaluate Algorithm Performance
- Review/Evaluate Developer's Documentation
  - Draft Part II Product Specification
  - DT&E Test Plan
  - Standards and Procedures Manual
  - Preliminary DT&E Test Procedures
  - Updated Computer Program Development Plan
- Qualify IV&V Tools
  - Conduct Checkout and Qualification
  - Conduct Required Reviews and Audits
  - Deliver and Install as Required
- Participate in Preliminary Design Review
- Participate in Critical Design Review
- Finalize IV&V Test Plan

The PO continues to monitor progress and reviews and approves products produced by both participants. The PO attends reviews, chairs working group meetings, institutes technical interchange meetings, attends design inspections, approves minutes, and assigns action items. Their major responsibility is to make sure the design is finalized by CDR. The PO must also make sure that all the supporting checkout and production equipment is available once the decision is made to go ahead with coding. They may wish to use an incremental development approach and hold several CDR's to preserve the schedule.

4.3.2 Concepts and Methods

Design verification is concerned with evaluating the software design in detail in order to confirm that it serves as an appropriate baseline for coding. The draft Part II Product Specifications are evaluaated to ensure their provisions are both consistent with the Part I Development Specifications and adequate to do the users processing job. The evaluation is directed towards answering the questions posed in Table 4-7. Evaluation tools and techniques employed to answer these questions are described in Appendix B and include simulation, prototypes, design languages, walkthrus, design inspections, process construction, and performance evaluation.

The IV&V agency should direct their efforts towards evaluating the following four areas during design verification:

- Technical Adequacy/Performance
- Modularity and Maintainability
- Timing and Sizing
- Support Equipment Availability

The primary objective of design verification is to confirm the technical adequacy of the design. The total software design must be expressed in writing, simulated, analyzed, and evaluated as to risk expected performance, cost, and reliability. The evaluation must consider performance capabilities, system and software architecture,

Table 4-7. Software Design Verification Checklist

• Have all software requirements been addressed in the design and is there traceability?
• Are all the equations, algorithms, and input/outputs correct?
• Is the data base fully defined and is its architecture (structure and access methods) fully compatible with the logical design?
• Are the specific module capabilities and their complex control and data linkages defined?
• Are the inter-module communications and interface rules established in the Part I Development Specification fully adhered to in the design?
• Is the design compatible with the hardware and software interfaces established in ICD's and the Part I Development Specifications?
• Does the design reflect the current version of the requirements (includes all ECP's)?
• Are there timing and sizing budgets established at the module level?
• Are the test procedures compatible with the design, test plan and Part I Development Specification test requirements?
• Do the individual designs fully realize overall requirements for performance, operation, growth, maintainability, etc?
• Is the design detailed enough to begin coding?
• Is there sufficient timing and sizing margin at CDR?

operational sequences, information flow, timing, scenario design and many other parameters. Some IV& V approaches that have proven successful in the past include:

- Use of design languages to document the design incrementally
- Use of discrete event architectural simulations to assist in making key design decisions relative to intermodule sequencing, control laws, communications processing and/or executive structure.

- Use of trial coding to confirm the performance or resource consumption of critical modules (identified during requirements verification) in a typical operating environment under nominal and stress conditions.

- Use of rederivation of key algorithms to assure optimality and to understand assumptions and approximations.

- Use of dimensional analysis to evaluate algorithms and data for completeness and compatibility.

Design verification activities must also ensure that the design is modular and maintainable. Software should be designed to accommodate change. The design should be evaluated to make sure the modularity rules (e.g., minimize intermodule communications using the Parnas information-hiding principle), testability and maintainability considerations are embedded within its structure. These provisions cannot be implemented as an afterthought. They must be an integral part of the design or else they will fail to be effective.

The next area of concentration for the IV&V agency is their timing and sizing analysis. The IV&V agency should continue to refine their estimates and compare them with those derived by the developer. The resulting budgets will be more realistic as a result.

The final area of concern is that of support equipment availability. The IV&V agency should assist the PO by monitoring the developer to ensure that the support software (e.g., compilers, simulators, etc.) and checkout equipment that is needed to start coding is available at the CDR. The IV&V agency must also police itself and assure that its tooling is available and qualified as well.

4.3.3 Products and Problems

The typical products and potential problems associated with the design verification activity are illustrated in Table 4-8.

4.4 PROGRAM VERIFICATION

Program verification is the V&V activity conducted to independently assure that the actual code that is developed is compliant with the technical description contained within the approved design specification. Program

Table 4-8. Software Design Verification Products and Problems

Products	Problems

<u>Developer</u>
• Test Plan
• Standards and Procedures Manual
• Draft Part II Product Specification
• Preliminary Test Procedures
• Users Manual
• Draft Version Description Document (VDD)
<u>Program Office</u>
• CRISP
<u>IV&V</u>
• Users Manual (Tools)
• Part II Product Specification (Tools)
• DT&E Test Report (Tools)
• VDD (Tools)
• Reports
    • Design Analysis
    • Timing and Sizing
• Discrepancy Reports
• Draft Test Procedures
	• Lack of Team Work and Petty Disputes
	    • Continual PO pressure to maintain interface and be constructive
	• Questionable Adequacy of Software Design
	    • Requirements volatile (premature baseline)
	    • Design not modular
	    • Design incompatible with machine selected
	    • Design architecture fails to accomodate use of existing software
	• No Preliminary User's Manual or Test Procedures Developed
	    • Failure to involve the user during design process
	    • Failure to make design testable

verification is that activity that ensures sanity, evaluates sequencing logic, file structuring, execution paths and limitations, and interfaces to name a few. This activity does not, however, evaluate the program's performance in a real or pseudo-real environment. That is the task of validation.

Program verification is a Full Scale Engineering Development Phase Activity. It takes the code as it is produced and compares it with the design specifications against which it was generated. It works with the object and source code. It is usually scoped to complement the developer's DT&E activities, not to duplicate them. Program verification is not a DT&E or a software integration activity. It may employ DT&E methods, but its aim is different. It is a separate and independent activity directed towards providing the PO with additional assurance that the code will properly realize the design. The tasks performed by the IV&V agency during this period are listed in Table 4-9. The output of this activity is code that fulfills its specifications.

Table 4-9. Program Verification Tasks

- Code Analysis (Source Level)
  - Design Compatibility /Traceability
  - Error-Prone Analysis
  - Execution Analysis
  - Static Analysis
  - Computer Resource Efficiency
- Machine Level Code Testing
  - Open-Loop Analysis
    - Unit
    - Module
  - Closed-Loop Analysis
    - Subsystem
    - System

    - Design Compliance
    - Stressed/Perturbed Functional Testing
  - Logical Testing
- Participate in Test Readiness Review

4.4.1 Responsibilities-Government and Contractors

The development contractor's responsibilities during the period starting with the CDR and ending with Final Qualification Testing (FQT) is to (1) code and checkout the CPC's, (2) integrate the CPC's into CPCI's, (3) conduct successful Preliminary Qualification Tests (PQT's) and Final Qualification Tests (FQT's) for all CPCI's, (4) support the conduct of formal audits, and (5) support the conduct of a Test Readiness Review. The developer starts with the approved design specifications and implements them. Implementation can be accomplished using a top-down (i.e., build-a-little and test-a-little), bottom-up or alternative methodology (e.g., hardest-out-first). Each CPC developed is tested stand-alone and in combination with other CPC's. Integration tests for the CPCI are then accomplished using regression, string, or other testing approaches. Finally, FQT's are conducted and audits are held. FQT's are formal tests of the integrated CPCI, performed by the contractor and witnessed by the PO, conducted to demonstrate that the CPCI fulfills the requirements of the Part I Development Specification. They differ from PQT's in the following areas:

- PQT's are normally much more detailed in terms of coverage.
- PQT's normally provide only minimal hardware/software interface testing.
- PQT's are normally conducted at the contractor site using simulated equipment and environments.

The IV&V agency's responsibility during this period is to independently test and evaluate the developer's product(s) using his own facilities and tools. The code is checked for errors, omissions, and incorrect translations using a variety of methods during production. The IV&V agency must do as much analysis as necessary to verify that the code correctly implements the design. The IV&V activity differs from the developer's DT&E tasks in the following areas:

- Program verification is conducted against the Part II Product Specification rather than the Part I Development Specification
- Program verification is usually less formal and less structured than either PQT or FQT

- Program verification is usually more stress oriented than PQT
- Program verification is conducted to discover and correct programming errors, not to confirm proper implementation (a major philosophical difference).

While program verification looks at design, validation may look at Part I requirements in addition to system specification needs. The guidebook seeks to clarify the distinction in roles for the reader in the next section.

The PO again monitors progress and reviews and approves products produced by both participants. The PO attends reviews, chairs working group meetings, institutes technical interchange meetings, resolves discrepancies, approves changes to specifications, approves minutes, and assigns action items. They conduct audits (both formal and informal) during this critical period to assess progress and confirm that the product that underwent test and that delivered are one in the same. They observe test conduct and analyze test results.

4.4.2 Concepts and Methods

Program verification is concerned with providing confirmation that the code fulfills the requirements of the Part II Product Specification. Confirmation is accomplished by addressing the questions listed in Table 4-10. Tools and techniques employed to answer the questions posed by the checklist are described in Appendix B. As one can observe from the Appendix, most so-called V&V tools and techniques address this activity. They have been developed in many cases to help perform unit, module, subsystem, and integration testing. These tools analyze the code in detail to determine whether there are errors present.

The IV&V agency should direct their efforts towards evaluating the following three areas during program verification:

- Technical Correctness
- Efficiency
- Technical Adequacy

Table 4-10. Program Verification Checklist

• Has every CPC been checked to determine whether it produces correct output for prescribed inputs?
• Are the arithmetic results correct for nominal conditions?
• Are the minimum and maximum inputs processed correctly?
• Are singularities and other conditional occurrences of data processed correctly?
• Are the subroutine calls properly formatted and has each been tested?
• Are the parameters dimensionally correct and is their calling sequence properly invoked?
• Is scaling proper to realize correct precision and desired results?
• Have all error conditions been processed correctly?
• Have all instructions and each branch been exercised at least once?
• Have the timing and resource allocations been properly mechanized?
• Is the task sequencing proper to mechanize the function in correct execution order?
• Is the compiler producing acceptable code?
• Are there any violations of agreed-upon programming practices?
• Is the users and program description documentation adequate?

The primary objective of program verification is technical correctness. The actual program code in its source and object form is evaluated against its design specification and discrepancies such as those listed below are identified for correction:

- Incorrect logic flow
- Inaccuracies in mathematical calculations
- Incompatible interfaces
- Improper use of instruction

Some IV&V approaches that have proven successful in identifying these and other errors in the past include:

- Use a verification approach that combines the virtues of functional, logical and path testing.
- Concentrate your effort on the interfaces and sequencing logic. Statistics show these areas to be very error-prone.
- Perform both static and dynamic execution analysis of the code. Static analysis will scrutinize the code and execution analysis will scrutinize the results.
- Use tools and approaches that allow for test repeatability and variable fidelity.

Program verification also addresses the efficiency problem. The program is continuously monitored as it is being developed to insure that timing and sizing budgets established during design are met. Detailed module timing analyses are conducted to identify CPC's that are marginal in processing data within prescribed time limits. Size is monitored. A key problem that typically causes size and timing growth is compiler inefficiency. The target computer code generator usually requires modifications to its optimization techniques even in the best of circumstances. The use of floating-point instructions in excess of what is thought to be an optimal mix for the intended application is another problem area.

The final area of concern is the technical adequacy of the code and related software products. Program verification ensures that the code is fully and correctly described in the Part II Product Specification which serves as as-built documentation. The Part II Product Specification should describe the program, not some lesser version of it. Program verification is also concerned with ensuring that the Users Manual is adequate. Lastly, program verification is concerned with assuring that the documentation adequately tracks the latest versions of the code.

4.4.3 Products and Problems

The typical products and potential problems associated with the program verification activity are illustrated in Table 4-11.

Table 4-11. Program Verification Products and Problems

Products	Problems

<u>Developer</u>
• Part II Product Specification
• Test Procedures
• DT&E Test Report
• Users Manual
• Version Description Document
<u>IV&V</u>
• Reports
    • Code Analysis
    • Open-Loop Analysis
    • Closed-Loop Analysis
• Discrepancy Reports
• IV&V Test Results Report
	• Questionable Adequacy of the Code
	    • Fails to adhere to standards and conventions
	    • Mechanization problems on the target computer
	    • Resource utilization problems and conflicts
	• Late Delivery of Hardware
	    • Impacts interface and integration testing
	    • Impacts logical testing
	    • Makes it hard to preserve schedule
	• Unreliable Hardware
	    • Causes software requirements to change (use software to compensate)
	    • Must rely on software test bed testing
	• Reliance on Nominal Test Results
	    • Insufficient-must do stress testing

4.5 SOFTWARE VALIDATION

Software validation is the V&V activity concerned with determining whether all software and system performance, interface, functional and test requirements are being satisfactorily fulfilled. Software validation is that activity that ensures that every requirement is adequately tested and that the software has been adequately shaken down from a system perspective. Unlkie program verification, validation seeks to evaluate the program's performance in a real or pseudo-real environment.

Software validation is a Full Scale Engineering Development Phase activity normally conducted somewhat in parallel with program verification. It takes the code as it is produced and compares it with the System Specification and Part I Development Specification requirements against which it was generated. It works with both the source and object code. It differs from program verification in purpose and in detail. Validation usually involves operational exercise of the code to assure that the requirements are met, while program verification usually involves the detailed analysis required to verify the design's proper implementation. In some instances, software validation activities overlap those conducted by the developer in the area of DT&E. The IV&V agency is tasked with providing a second opinion on the software's ability to perform. The IV&V agency will test those critical functions identified during system specification and software requirements verification to provide the evidence he needs to confirm the software's capabilities. If the entire program is critical, the IV&V will run a totally independent DT&E to qualify it from their vantage point. Typical validation tasks conducted by the IV&V agency during this period are listed in Table 4-12. The output of this activity is code that fulfills system level requirements.

4.5.1 Responsibilities - Government and Contractors

The development contractor's responsibilities during the period starting with the FQT and ending at the System Readiness Review is to (1) integrate the CPCI's with other CPCI's and the hardware, (2) conduct system level tests, and (3) support the conduct of formal reviews and audits and IOT&E. The developer starts with CPCI's and CI's qualified by FQT/FCA/PCA. He integrates them together and tests the composite

Table 4-12. Validation Tasks

- Implement V&V Test Plan, Formal and Controlled Testing
  - Verify <u>All</u> Software <u>And</u> System Requirements Are Met
    - Performance
    - Functional
    - Interfaces Both Internal and External
  - Stress/Failure Performance Testing
- Review Final Developer's Documentation, All Specifications and User's Manuals
  - Complete
  - Accurate/Compatible With Delivered Programs
  - Compliant With Standards
- Development Test Results Evaluation
  - Complete/Consistent With Test Plan
  - Traceable to Requirements
  - Comparison With IV&V Results
- Participate in System Readiness Review, FCA and PCA

system in accordance with the provisions of the TEMP. In some instances, the system is transitioned to a Government team which conducts IOT&E of the integrated system before it is deployed.

The IV&V agency's responsibilities begin earlier in the life cycle. They test and evaluate the code that was identified as critical in parallel with both its code verification and the developer's DT&E activities. Their job is to provide feedback early enough so that problems identified can be corrected without costly schedule impacts. The IV&V agency accomplishes its job by providing independently derived test results against which the developer's results can be compared. The IV&V agency also actively participates in the FQT and the System Readiness Review. The formal results of the IV&V agency's test and evaluation efforts are presented at these reviews.

The PO continues to monitor progress and review and approve products produced by both participants. The PO participates and witnesses test conduct and analyzes test results. They chair working group meetings where both the developer and the IV&V agency present the results of their testing. They resolve problems and act as the arbitrator for disputes. They attend reviews, chair technical interchange meetings, approve changes to the specifications, approve minutes, and assign action items.

4.5.2 Concepts and Methods

Programs are validated to confirm that they perform in accordance with their system and software requirements. Confirmation is accomplished by executing the completed code in a realistic environment according to the following three stage approach:

- CPCI Testing
- Integrated CPCI Testing
- System Testing

CPCI testing is that formal testing conducted to confirm that each and every requirement of the approved Part I Development Specification has been fulfilled. CPCI testing is accomplished by the developer, witnessed by the PO and independently evaluated by the IV&V agency (if warranted). It involves both PQT and FQT. It can be achieved incrementally in either a top-down or bottom-up fashion. It uses approved DT&E procedures which are compatible with the test plan approved for demonstrating the Part I Development Specification requirements. A checklist for the conduct of CPCI testing is illustrated in Table 4-13.

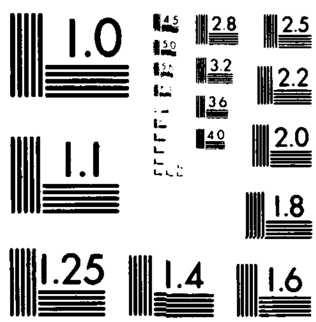
The following problems should be addressed by all parties that participate in CPCI testing:

- Designing an effective set of test cases.
- Creating an efficient test environment.
- Managing test data.
- Knowing when to stop testing.

AD-A090 966 TRW DEFENSE AND SPACE SYSTEMS GROUP REDONDO BEACH CA F/G 9/2
AIRBORNE SYSTEMS SOFTWARE ACQUISITION ENGINEERING GUIDEBOOK FOR--ETC(U)
SEP 78 D J REIFER F33657-76-C-0677
UNCLASSIFIED TRW-30323-6009-TU-00 ASD-TR-79-5028 NL

2 of 2
AD A
090966

END
DATE
FILMED
12-80
DTIC



MICROCOPY RESOLUTION TEST CHART

Table 4-13. CPCI Testing Checklist

• Are all inputs accepted and all outputs produced?
• Does the mechanization of algorithms and models fulfill the prescribed requirements?
• Can the function being performed by the module be exercised at the extremes of the range of input variables?
• Are the initialization provisions properly implemented?
• Are the error handling provisions properly mechanized and has every error condition been tested?
• Are the relationships between the program and the CPCI clear?

Methods that have been used effectively to attack these problems include:

- Designing test cases against established test criteria similar to those listed in Table 4-14.
- Designing test cases that exercise software capabilities, not features.
- Using test tools effectively to create an efficient diagnostic environment.
- Creating a test data base that relates each test to its requirements and manages test cases and test results.
- Setting pre-defined, realistic goals against which test accomplishment can be measured.

Table 4-14. Example Test Criteria

• Programmer judgement
• Execution of all program statements
• Execution of all program branches
• Dividing program paths into equivalence classes and executing at least one path from each class
• Execution of randomly-selected test data
• Execution of all legal program paths
• Stress test at boundaries

The IV&V agency's involvement in CPCI testing is dependent on their assessment of criticality. For non-critical cases, they should participate as an independent reviewer of the developer's products. Test documentation should be revived with a concentration on procedures and results. The procedures should be reviewed to ensure:

- The test procedure tests the program and not a simpler variation of it.
- There is positive feedback in every test procedure.
- The results of the test procedure are not only predictable, but predicted.
- Test results meet all acceptance criteria.

Test results should be evaluated against expectations, previous results, and requirements. Results should be further examined to determine if the test objectives established have been realized. Criteria established in the test plan and Part I Development Specification serve as guides to this determination.

The IV&V agency should independently test those CPCI's designated as critical. Test plans and procedures should be developed to define what tests will be conducted to evaluate the program. Care should be exercised to ensure that the test program is independent of the developer. The major differences between the IV&V and the developers test program are as follows:

- The developer's test program is much more formal. PQT's and FQT's are conducted for each CPCI. The IV&V agency's test program is usually less structured.
- The IV&V agency's concentration is more test oriented because that is their one and only job.
- The developer stresses functional testing at the expense of logical, path and stress testing. The IV&V agency normally gives equal attention to all four techniques.

Integrated CPCI testing is accomplished to validate the overall operation of the data system as an integrated entity in a pseudo-operational environment. The CPCI's are integrated together and with the hardware

and tested to ensure that the provisions of the system or system segment specification are fulfilled. The common myth that the developer's software organization's job ends with validated CPCI's is disspelled. The software developer participates as an essential member of the test team. As the CPCI's and CI's are integrated and tested, software personnel accomplish the following tasks:

- Review test results and identify problems.
- Review problem solution approaches for potential software impacts.
- Modify qualified CPCI's.
- Retest modified CPCI's.
- Modify Part I Development Specifications and related documentation.
- Advise the Test Director of the problems and pitfalls associated with solving all the world's problems through software.

The IV&V agency supports integrated CPCI testing and contributes directly as a member of the team. They provide independent impact assessments and reverify and revalidate modified CPCI's as required by a needs assessment.

Finally, the integrated system is validated in an operational environment and turnover and transitioning is accomplished. System validation is accomplished to ensure that all the provisions of the System Specification are fulfilled. The following planning documents play an important role in system testing:

- Test and Evaluation Objectives Annex (TEOA)
- Test and Evaluation Master Plan (TEMP)
- Computer Resources Integrated Support Plan (CRISP)
- Program Management Responsibility Transfer (PMRT) Agreements
- Turnover Agreements

System testing is normally conducted by a government team with assistance provided as needed by the developing contractor. The IV&V

agency may participate as a member of the team.

4.5.3 Products and Problems

The typical products and potential problems associated with the validation activity are illustrated in Table 4-15.

Table 4-15. Software Validation Products and Problems

Products	Problems

PO
• System Test Procedures
• System Test Reports
IV&V
• Reports
    • Critical function identification
    • Test evaluation
    • Post mortem
• Discrepancy Reports
• Test Procedures
• IV&V Test Report
	• Insufficient Time Allocated to System Test
	    • System prematurely accepted and fielded
	    • Incremental delivery concept useful
	• Limited Software Manpower Committed to Support System Test
	    • Hardware and system problems are usually corrected using software
	• Integration Disputes Cause Friction
	    • Continual PO pressure to act as a team and avoid finger pointing
	• Unreliable Hardware
	    • Backup facilities needed to preserve schedule

4.6 CERTIFICATION

Certification is the activity concerned with substantiating that enough evidence has been obtained to state with near certainty that the acquired system and its attendant software will satisfy the user's documented need. Certification is administrative in nature. It is the process leading to approval of the product.

There is no formal policy governing certification. It is an optional activity conducted at the discretion of the PO. It is accomplished in many different ways by many different people. Some systems are certified by decree. Others are certified after an extensive Operational Test and Evaluation (OT&E). Some use formal procedures, while others don't. Some systems are certified by the using command, others by an Air Force test team. Needless to say, it seems that individual project circumstances govern the who's and how's of certification.

This section is based upon the philosophy that certification should be accomplished formally with wide latitude. The latitude is needed to cope with project peculiarities and needs. For example, certification procedures for an aircraft radar system upgrade need not be as stringent as those for an entirely new fighter aircraft. This section was also written to suggest one way of doing business. It should not be interpreted as recommending either policy or prescribed practice. That recommendation can only be made by the government team responsible for the system.

4.6.1 Responsibilities - Government and Contractors

The responsibility for certification is the government's and theirs alone. It should not be delegated to contractors. Contractors can support certification by providing information. They should not certify a system.

Certification can be accomplished in many ways depending upon the unique needs of the system. This discussion will treat the following two cases to illustrate the suggested approach:

- A new aircraft employing a new computer controlled avionics suite.
- An avionics system upgrade.

For a new aircraft, certification is optimally accomplished after IOT&E, but before PMRT. PO personnel working with the Air Force test team and representatives from the implementing, supporting and using commands, under the auspices of the Computer Resources Working Group (CRWG), review the audit and test results (i.e., DT&E, IOT&E, Verification and Validation) to confirm the system's operational effectiveness and suitability (including compatibility, interoperability, reliability, maintainability, and logistics and training requirements) from a computer resources point of view. The team also determines if the provisions of the inter-agency coordinating agreements (e.g., PMRT, turnover, etc.) have been adhered to. Each and every item in the governing documents is assessed. The results of the assessment are used to determine computer resource deficiencies that either have to be corrected before PMRT or listed in the PMRT or turnover certificate. Each organization representative should affix their signature to the certificate prior to turnover and transfer. The program should be bonded (tapes placed in sealed containers) and the related documentation marked appropriately.

For a system upgrade, certification is optimally accomplished after DT&E. PO personnel working with the Air Force test team and representatives of the using command review the test results (i.e., DT&E, Verification and Validation) to confirm that the objectives stated in the Test and Evaluation Objectives Annex (TEOA) and the TEMP have been fulfilled. The review results are used to determine computer resources deficiencies that either have to be corrected before the system is declared operational or listed for incorporation in the next program update. A certificate certifying the system is then issued with signatures of government participants affixed. The program is bonded and the related documentation marked appropriately.

4.6.2 Concepts and Methods

Certification is an administrative activity that should be planned and executed in an orderly manner. The primary planning documents that impact certification procedures are those that detail inter-agency

coordinating agreements. These documents could include:

- Program Management Directive (PMD)
- Program Management Plan (PMP)
- Test and Evaluation Objectives Annex (TEOA)
- Test and Evaluation Master Plan (TEMP)
- Integrated Logistics Support Plan
- Computer Resources Integrated Support Plan (CRISP)
- PMRT Agreements
- Turnover Agreements
- Memorandums of Agreement (MOA)
- Operational/Support Configuration Management Procedures (O/S CMP)
- Interface Control Documents

These documents delineate the following:

- Organizational roles and responsibilities.
- Scheduled delivery dates of applicable computer resources.
- Applicable documentation and technical publications.
- Facility, support and training requirements.
- Budgetary and manpower requirements.
- Configuration management and quality assurance provisions.

Certification is accomplished by having the government team review audit and test results in order to make a determination and finding. A checklist for such a review is illustrated in Table 4-16. The result of the determination and finding is an action plan to be used for fielding the system. After all, the primary objective of the entire development effort is to provide the user (e.g., pilot, astronaut, etc.) with a system that he can use to accomplish the desired mission (e.g., close air support, etc.).

4.6.3 Products and Problems

The typical products and potential problems associated with certification are listed in Table 4-17.

Table 4-16. Certification Checklist

• Have budgetary needs been identified and appropriate funding allocated?
• Are current versions of the program and its documentation compatible and have all ECP's been incorporated?
• Have configuration management procedures been established and are they adequate for the potential change workload?
• Is the support facility available and is it adequately manned by trained personnel?
• Have all specified design and performance requirements been successfully demonstrated by DT&E?
• Have the system level test objectives in the TEOA been successfully demonstrated by OT&E?
• Have all action items initiated as a result of reviews, audits, working group and/or technical interchange meetings been closed and are the results of the closure satisfactory?
• Have critical functions been subjected to an independent verification and validation and have all identified discrepancies been satisfactorily resolved?
• Have all inter-agency agreements been honored?
• Have adequate procedural mechanisms been established to handle future maintenance and logistics needs?

Table 4-17. Certification Products and Problems

Products	Problems

<u>Developer</u>
• Supporting Information
<u>IV&V</u>
• Supporting Information
<u>Government</u>
• Reports
    • Test evaluation
    • Certification
• OT&E Evaluation Report
• Transfer Certificate
• Turnover Certificate
	• Inordinate Number of Discrepancies Found
	    • Lack of maturity
	    • Indication that additional testing warranted
	• Insufficient Government Manpower
	    • Off-load non-critical tasks to contractors
	    • Prioritize
	• External Pressure to Field System Prematurely
	    • Commitment dates frozen
	    • Difficult to say "no"
	    • Cost/benefits of field retrofit must be evaluated
	• Maintenance Facilities Late
	    • Good planning can prevent

APPENDIX A

Glossary of Key Terms

1. Certification: The formal administrative procedures established to substantiate that enough evidence has been obtained to state with near certainty that the performance of the acquired system and its attendant software will satisfy the user's documented need.

2. Computer Data: Basic elements of information used by computer equipment in responding to a computer program. Data operated on, produced by, or otherwise used by a computer program (AFR 800-14, Volume I/AFSC Sup. 1).

3. Computer Program: A series of instructions or statements in a form acceptable to an electronic computer, designed to cause the computer to execute an operation or operations (AFR 800-14, Volume I).

4. Computer Resources: The totality of computer equipment, computer programs, associated documentation, contractual services, personnel, and supplies. Microprocessors, microcomputers and firmware are considered computer resources (AFR 800-14, Volume I).

5. Evaluation: The review and analysis of qualitative and/or quantitative data obtained from design review, hardware inspection, testing and/or operational usage of equipment (AFR 800-14).

6. Firmware: Most commonly defined as computer programs and computer data at the microprogram level. Also applies to any level of executable computer programs and computer data that cannot be readily modified under program control, that is, read only (AFR 800-14, Volume I/AFSC Sup. 1).

7. Life Cycle Cost: The total cost of an item or system over its full life. It includes the cost of development, production, ownership (operation, maintenance, support, etc.) and, where applicable, disposal (AFR 800-14).

8. <u>Operational Effectiveness:</u> How well the system performs its intended mission in its intended environment, exclusive of system support considerations. Survivability, compatibility, and interoperability may be considerations in evaluating effectiveness (AFR 800-14).

9. <u>Operational Suitability:</u> How well the system performs its intended mission when operated and maintained by military personnel in the field. This normally includes capability, availability, reliability, maintainability, logistics supportability, training requirements, and an assessment of operating and support cost characteristics (AFR 800-14).

10. <u>Software Engineering:</u> Science of design, development, implementation, test, evaluation, and maintenance of computer software over its life cycle (DODD 5000.29).

11. <u>Test:</u> Any program or procedure which is designed to obtain, verify, or provide data for the evaluation of: research and development other than laboratory experiments; progress in accomplishing development objectives; or performance and operational capability of systems, subsystems, components, and equipment item (AFR 800-14).

12. <u>Transfer:</u> That point in time when the designated Supporting Command accepts program management responsibilities from the Implementing Command. This includes logistic support and related engineering and procurement responsibilities (AFR 800-14).

13. <u>Turnover:</u> That point in time when the Operating Command formally accepts responsibility from the Implementing Command for the operation and maintenance of the system equipment, or computer program acquired (AFR 800-14).

14. <u>Validation (of Computer Programs):</u> The evaluation, integration and test activities conducted at the system level to ensure that the finally developed software satisfies applicable requirements set down as performance and design criteria in the system specification and/or Part I Development Specification (AFR 800-14, Volume I/AFSC Sup. 1).

15. <u>Verification (of Computer Programs)</u>: **The iterative process of determining whether the product of each step of the Computer Program Configuration Item (CPCI) development and change process fulfills all requirements levied by the previous step (AFR 800-14, Volume I/AFSC Sup. 1).**

APPENDIX B

TOOLS AND TECHNIQUES SURVEY[†]

Software tools and techniques can serve as powerful aids in Verification, Validation and Certification. Tools are typically computer programs which are used to automate tedious, error-prone, manual processes and to produce information that cannot be economically produced manually. Tools can increase both productivity and quality. Techniques are guides to disciplined action. Techniques prescribe technical methods and procedures for accomplishing a desired aim in an effective manner. Techniques are formulated on the basis of usually postive experience.

This appendix categorizes tools and techniques and provides a brief description of each. Comparisons between tools within the same category can be found in several of the documents referenced in Appendix C of this Guidebook. The forty-six tools and twenty-nine techniques listed in Table B-1 are grouped according to the Verification, Validation or Certification function they support in Table B-2.

This appendix neither claims to be a complete listing nor an endorsement of tools and techniques. For example, it does not list tools and techniques useful in software development or quality assurance. Each tool and technique is briefly described alphabetically in the glossary that follows.

1. <u>Accuracy Study Analyzer</u>. A computer program used to perform calculations to assist in determining if program variables are computed with required accuracy.

2. <u>Algorithm Evaluation Test</u>. A technique employed to evaluate critical algorithm tradeoffs (i.e., speed vs. size vs. precision) before the design is finalized. Algorithms are trial-coded and exercised in a simulated environment to ensure mission requirements are satisfied.

[†]Summary descriptions of development tools can be found in Reifer and Trattner, "A Glossary of Software Tools and Techniques," <u>Computer</u>, July 1977; and of quality assurance tools in Reifer, "Software Quality Assurance Tools and Techniques," <u>Software Quality Management</u>, Petrocelli, 1978.

Table B-1. List of Tools and Techniques

Tools		
Accuracy Study Analyzer	Editor	Requirements Tracer
Assembler	Engineering (Scientific) Simulations	Restructuring Program
Automated Test Generator	Environmental Simulator	SNAP Generator
Comparator	Flowcharter	Software Monitor
Compiler	Hardware Monitor	Standards Enforcer
Compiler Validation System	Instruction-Level Simulator	Statement-Level Simulator
Consistency Checker	Instruction Trace	Static Analyzer
Cross-Assembler	Interface Checker	Test Beds
Cross-Reference Program	Interrupt Analyzer	Test Drivers, Scripts, Generator
Data Analyzer	Logic/Equation Generator	Test-Result Processor
Decision Tables	Overlay Program	Timing Analyzer
Decompiler	Path Analyzer	Trace
Design Language Processor	Program Sequencer	Units Consistency Analyzer
Diagnostic/Debug Aids	Relocatable Loader	Workload Analysis Aids
Driver	Requirements Language Processor	
Dynamic Analyzer		
Dynamic Simulator		

Techniques		
Algorithm Evaluation Test	Functional Testing	Standardization
Analytical Modeling	Logical Testing	Static Analysis
Capability Matrices	Modular Programming	Stress Testing
Code Inspection	Path Testing	Structured Programming
Correctness Proofs	Performance Evaluation	Symbolic Execution
Design Inspection	Post Functional Analysis	System Simulations
Emulation	Process Construction	Top-Down Programming
Equivalence Classes	Production Libraries	Walkthrus
Error-Prone Analysis	Prototyping	
Execution Analysis	Simulation	
Flight Tests		

Table B-2. Verification, Validation and Certification Relationships

Tool and Technique	Verification				Validation	Certification
	System Requirements Verification	Requirements Verification	Design Verification	Code Verification		
1. Accuracy Study Analyzer				X		
2. Algorithm Evaluation Test			X			
3. Analytical Modeling		X	X			
4. Assembler				X	X	
5. Automated Test Generator				X	X	
6. Capability Matrices	X	X				
7. Code Inspection				X		
8. Comparator				X	X	
9. Compiler				X	X	
10. Compiler Validation System					X	
11. Consistency Checker	X	X	X			
12. Correctness Proofs			X	X		
13. Cross-Assembler				X	X	
14. Cross-Reference Program				X		
15. Data Analyzer				X		
16. Decision Tables		X	X			
17. Decompiler				X		
18. Design Inspection			X			
19. Design Language Processor			X			
20. Diagnostics/Debug Aids				X	X	
21. Driver				X	X	
22. Dynamic Analyzer				X		
23. Dynamic Simulator				X	X	
24. Editor				X		
25. Emulation				X	X	
26. Engineering Simulations	X	X	X			
27. Environment Simulator				X	X	
28. Equivalence Classes				X	X	
29. Error-Prone Analysis				X		
30. Execution Analysis				X	X	X
31. Flight Tests						X
32. Functional Testing				X	X	X
33. Flowcharter			X	X	X	
34. Hardware Monitor				X	X	
35. Instruction Simulator				X	X	
36. Instruction Trace				X		
37. Interface Checker				X		
38. Interrupt Analyzer				X		
39. Logical Testing				X	X	
40. Logic/Equation Generator				X		
41. Modular Programming				X	X	
42. Overlay Program				X		
43. Path Analyzer				X		
44. Path Testing				X		
45. Performance Evaluation	X	X	X	X	X	X
46. Post Functional Analysis				X		
47. Process Construction			X			
48. Production Libraries			X	X	X	
49. Program Sequencer				X		
50. Prototyping	X	X	X			
51. Relocatable Loader				X		
52. Requirements Processor	X	X				
53. Requirements Tracer		X	X			
54. Restructuring Program				X		
55. Simulation	X	X	X	X	X	X
56. SNAP Generator				X		
57. Software Monitor				X	X	
58. Standardization		X	X	X	X	X
59. Standards Enforcer				X		
60. Statement Simulator				X		
61. Static Analysis				X		
62. Static Analyzer				X		
63. Stress Testing				X	X	
64. Structured Programming				X	X	
65. Symbolic Execution				X		
66. System Simulations	X	X				
67. Test Beds				X	X	X
68. Test Drivers, Scripts, Generators				X	X	X
69. Test-Result Processor				X	X	
70. Timing Analyzer				X		
71. Top-Down Programming		X	X			
72. Trace				X		
73. Units Consistency Analyzer				X		
74. Walkthrus			X	X		
75. Workload Aids	X	X				

3. Analytical Modeling. A technique used to express mathematically (usually by a set of equations) a representation of some real problem. Such models are valuable for abstracting the essence of the subject of inquiry. Because equations describing complex systems tend to become complicated and often impossible to formulate, it is usually necessary to make simplifying assumptions, which may distort accuracy.

4. Assembler. A computer program that translates source instructions in mnemonic form into a machine-language program. Assembler diagnostics/debug aids are useful in detecting errors in the code.

5. Automated Test Generator. A computer program that accepts inputs specifying a test scenario in some special language, generates the exact computer inputs, and determines the expected results. The NASA-developed Automated Test Data Generator serves as an example. ATDG takes identified code segments, defines a patch through the module under test, and generates input data required to execute the selected paths.

6. Capability Matrices. A technique used to trace functions or their interfaces versus requirements. The technique is also known as N^2 charting.

7. Code Inspection. A disciplined technique used for inspecting the code and identifying discrepancies. Participants have well defined roles and criteria for evaluating the code. If errors are identified, deficiency reports are generated. Follow-up procedures are used to ensure that the errors have been corrected.

8. Comparator. A computer program used to compare two versions of the same computer program under test to establish identical configuration or to specifically identify changes in the source coding between the two versions.

9. Compiler. A computer program that either transforms a higher order language source program into an assembly language form (i.e., mnemonics for machine code) for subsequent assembly to machine language translation by the assembler, or that transforms directly the higher order language program into an equivalent machine-language program. A compiler contains a syntax analyzer, which parses source program text, and a code generator, which emits machine-level instructions for the target computer in optimized form. Compiler diagnostics and utilities are primary verification and validation tools.

10. Compiler Validation System. Computer programs used to ensure that compilers written meet their language specification. The best known system is that used by the Federal COBOL Compiler Testing Service to validate COBOL compilers for the government.

11. Consistency Checker. A computer program used to determine (1) if requirements and/or designs are consistent with each other and their data base, and (2) if they are complete. The TRW-developed Design Assertion Consistency Checker (DACC) is an example. DACC has been used to detect inconsistencies between assertions about the nature of inputs and outputs of the various elements of a software design.

12. Correctness Proofs. Automated verification systems exist which allow the analyst to prove the correctness of small programs by means similar to those used in proving mathematical theorems. Axioms and theorems derived are used to establish the validity of program assertions and to provide a fundamental understanding of how the program operates. Several approaches to proofchecking are being pursued. One approach uses symbolic execution to demonstrate program correctness a posteriori. Another approach proves correctness a priori. Interactive systems have been developed to implement both approaches.

13. Cross-Assembler. A computer program that accepts symbolic instruction mnemonics for a selected target computer and generates target-computer machine code while hosted on another computer. A cross-assembler thus allows code written for one computer to be assembled on another.

14. Cross-Reference Program. A group of computer programs that provide cross-reference information on system components. For example, programs can be cross-referenced with other programs, macros, parameter names, etc. This capability is useful in problem-solving and testing to assess impact of changes to one area or another.

15. Data Analyzer. A computer program that examines source data definitions and analyzes data relationships, data structures, formats and usage for consistency, validation and storage utilization. Errors dealing with improper specification, misuse and non-use of data and conflicts in data are identified.

16. Decision Tables. A mechanism used to represent information on program conditions, rules and actions in tabular form that can be automatically translated to executable code by a processor. Decision tables are a tabular representation of the design which can be used to clarify the control flows of decision alternatives by presenting the information in a concise and understandable format.

17. Decompiler. A computer program that accepts as data a program written in machine-level source language and produces as output the higher-level, problem-oriented, target language.

18. Design Inspection. A disciplined technique used for inspecting the design and identifying discrepancies. Participants have well-defined roles and criteria for evaluating the design. If errors are identified, the design is reworked. Follow-up procedures are used to ensure that the errors have been corrected.

19. Design Language Processor. A computer program used to provide an understandable representation of the software design as it evolves. These programs allow design to be constructed and expanded in a hierarchical fashion. They document the design and the decisions that led to it. The Boeing-developed Design Expression and Confirmation Aid (DECA) is an example. DECA is used in conjunction with a top-down dominated design methodology to organize, validate, and produce a design document.

20. Diagnostics/Debug Aids. Compile-and execution-time checkout and debug capabilities help identify and isolate program errors. They usually include commands or directives such as DUMP, TRACE, MODIFY CONTENTS, BREAKPOINT, etc.

21. Driver. A driver can perform one of two functions: control an external hardware device or control the execution of other programs.

22. Dynamic Analyzer. A computer program used to provide information about the execution characteristics of the source code. This type of tool instruments the source code by generating and inserting counters at strategic points to provide measures of test effectiveness. They monitor and record the execution dynamics and provide data that details how thoroughly the source code has been exercised. Many dynamic analyzers have been developed. Comparison analysis of several analyzers and user feedback with such systems has been published.

23. Dynamic Simulator. A computer program used to checkout a program in a simulated environment comparable to that in which it will reside. Closed-loop effects between computer and environmental models are gained when the various models respond to inputs and outputs. The simulator allows the environment to be stabilized at a specific configuration for any number of runs required to observe, diagnose, and resolve problems in the operational program. The dynamic simulator used at the Naval Weapons Center at China Lake, California, for update and test of operational flight software for the A-7 aircraft is an example.

24. Editor. A computer program used to analyze source programs for coding errors and to extract information that can be used for checking relationships between sections of code. The Editor will scan source code and detect violations to specific programming practices and standards, construct an extensive cross-reference list of all labels, variables, and constants, and check for prescribed program formats.

25. Emulation. Technique of microprogramming one computing system to execute computer programs written for another system with the register-level, bit-level, and memory-cycle instruction timing accuracy of the target machines.

26. Engineering (Scientific) Simulations. Engineering simulations are used to study system characteristics, test algorithms, and provide data that act as a standard for testing. These programs generally simulate subsystems at varying degrees of complexity, depending on the subsystem being studied or the use made of the simulation. They generally consist of a set of modules, each of which is assigned a specific simulation function and is designed with well-defined inputs and outputs and precise interfaces. Each module performs an assigned simulation function to vary the method, speed of computation, accuracy, complexity, etc. The structure can encompass all basic simulation capabilities for simulation of continuous and discrete systems. The Aerospace Corporation-developed Generalized Trajectory Simulator (GTS) serves as an example. GTS has been used to provide closed-loop simulations of the flight of various launch vehicles from liftoff to payload release in six dimensions. These closed-loop simulations use models of the launch vehicle, the propulsion system, and the environment in conjunction with flight equations executing on real or simulated target computers to validate the performance of flight software.

27. Environment Simulator. A computer program used to permit testing of operational programs on a host computer. The operational programs run under simulated conditions as if they were operating within the real-time control program of a machine to which all of the devices constituting the ultimate system are attached. The simulator program contains expansions of all control program macros that modify the entry block and other working storage in the same manner as the macros in the actual control program.

28. Equivalence Classes. A technique used to automatically identify a complete set of test cases for a program. The set is interpreted in terms of inequalities involving program variables that define a set of conditions necessary for the particular program flow to actually occur.

29. Error-Prone Analysis. A technique employed during code analysis to identify these areas of the program which required continuous correction and change. These areas can either be reworked or subjected to an extensive test effort.

30. Execution Analysis. A technique employed during test to investigate program behavior errors and to identify areas in the code that were either untested or not fully tested. The program is executed and statistics are collected. The results and the statistics are then analyzed to insure that each interface, functional, and test requirement has been correctly mechanized by the code.

31. Flight Tests. A technique used to demonstrate total system performance under realistic operational conditions.

32. Functional Testing. A technique used to demonstrate that software performs its specifications satisfactorily under normal operating conditions computing nominally correct output values from nominal input values.

33. Flowcharter. A computer program used to show in detail the logical structure of a computer program. The flow is determined from the actual operations as specified by the executable statements, not from comments. The flowcharts generated can be compared to flowcharts provided in the computer program specification to show discrepancies and illuminate differences. Several flowcharters such as AUTOFLOW and FLOWGEN are commercially available.

34. Hardware Monitor. A unit that obtains signals from a [illegible] computer through probes attached directly to the computer's circuitry. The signals obtained are fed to counters and timers and are recorded. These data are then reduced to provide information about system and/or program performance (CPU activity, channel utilization, etc.)

35. Instruction-Level Simulator. A computer program used to simulate the execution characteristics of a target computer at the machine level using a sequence of instructions of a host computer. The instruction simulator provides bit-for-bit fidelity with the results that would be produced by the target computer following the same operations and initial conditions. Instruction simulators are a major tool used in the Verification and Validation of flight software.

36. Instruction Trace. A computer program used to record every instance in which a certain class of operations occurs and triggers event-driven data collection. In some cases, this creates a complete timed record of events occurring during program execution.

37. Interface Checker. A computer program used to automatically check the range and limits of variables as well as the scaling of source programs to assure format compliance with interface and control documents.

38. Interrupt Analyzer. A computer program that determines potential conflicts to a system as a result of the occurrence of an interrupt.

39. Logical Testing. A technique employed to confirm that the code performs its computation correctly. Items validated by logical testing include arithmetic (i.e., accuracy, precision, etc.), error handling, initialization, interfaces, and timing.

40. Logic/Equation Generator. A computer program used to automatically reconstruct equations forming the basis of a program and to flowchart assembly language programs. One such program translates assembly language instructions into a machine-independent microprogramming language and builds the microprogramming statements into a network in which the flow of control is analyzed and equations are reconstructed.

41. Modular Programming. The technique of producing relatively small, easily interchangeable computer rotuines which meet certain standarized interface requirements. This technique makes it easier to develop and verify completed computer programs. Modularity is accomplished by breaking the program into limited line-segments that perform complete functions and are therefore completely understandable in themselves. Aids that help implement these techniques are standards and procedures.

42. Overlay Program. A computer program that allows specific system components (load modules, core, data base, etc.) to be modified during execution. In the case of modules, a program with an error can be replaced in core without bringing the system down and starting it up again. System parameters that effect performance can be varied during execution to compare various priority and timing schemes, etc.

43. Path Analyzer. A computer program that traces through all paths being exercised during test and generates statistics on source code statement usage and timing. A primary use of such programs is test case design.

44. Path Testing. A technique used to confirm that certain test-effectiveness measures based on the program's control topology have been realized. The technique ensures that a sufficient number of statements, branch paths and subroutine calls have been exercised during program execution. It also helps identify a complete set of test cases for the program.

45. Performance Evaluation. Techniques used to measure and predict performance of alternate system (both hardware and software) configurations over time. Measurement is accomplished using tools such as software and hardware monitors. Prediction is accomplished using simulation or workload evaluation tools like kernals, benchmarks, or synthetic programs.

46. Post Functional Analysis: A technique employed after completion of functional testing to identify functionally weak areas in the program. The recorded test results are analyzed and the quality of the final product is determined.

47. Process Construction. A technique used to combine and link independently coded modules into a run-time process. These include linkages to the operating system. The technique allows for rapid reconfiguration, based on stimuli from the run-time environment, of a software system to reflect changes made to a number of its modules. Specific computer programs are available that serve as aids to implementation. These include special-purpose editors and control programs.

48. Production Libraries. A technique used to provide constantly up-to-date representations of the computer programs and test data in both computer- and human-readable forms. The current status and past history of all code generated are also maintained. Specific library programs are available to serve as aids to implementation.

49. Program Sequencer. A computer program that forces execution of all possible instructions and branches within a program to determine program flow, to execute seldom-used branches, and to assist in the verification of proper program operations. The aid is often used with an instruction simulator.

50. Prototyping. A technique used where a quick and dirty, low cost version of the system is built to verify its design concepts and reduce the risk. Prototypes are either thrown away or serve as the core model for the production version (used to look at the impact of changes).

51. Relocatable Loader. A computer program that enables external references of symbols among different assemblies as well as the assignment of absolute addresses to relocatable strings of code. This program provides diagnostics on assembly overlap, unsatisfied external references, and multiple defined external symbols.

52. Requirements Language Processor. A computer program used to provide a succinct and unambiguous specification of the system based on computer requirements. More precisely, it allows requirements to be communicated and translated in a hierarchical manner. The most widely published of these is the Problem Statement Language (PSL) processor developed at the University of Michigan. Other requirements language processors are available.

53. Requirements Tracer. A computer program used to provide traceability from requirements through design and implementation of the software products. Traceability is characterized to the extent that each requirement specified is tracked in each succeeding level of documentation produced. The THREADS requirements tracer developed by Computer Sciences Corporation serves as an example.

54. Restructuring Program. A computer program that takes an unstructured source-language program as input and creates an equivalent structured replacement. The "structuring engine" developed by Caine, Farber and Gordon, Inc., converts non-structured Fortran programs into an equivalent structured form.

55. Simulation. A technique used to model and vary the characteristics of systems over time in order to predict performance, check understanding, determine the impact of change, or obtain information on system capacity. Simulations are usually aimed at increasing understanding of interactions and alternatives in complex systems.

56. SNAP Generator. A computer program used to provide program locations that are relative to program labels. A SNAP generator is typically used to present a picture of a selected portion of memory.

57. Software Monitor. A computer program that provides detailed statistics about system performance. Because software monitors reside in memory, they have access to all the tables the system maintains. Therefore, they can examine such things as core usage, queue lengths, and individual program operation to help measure performance.

58. Standardization. A technique used to create an authoritative model against which products and/or procedures can be compared in order to determine their relative quality. Software items for which standards should be established include documentation, languages, designs, and structured programming.

59. Standards Enforcer. Standards consist of procedures, rules, and conventions used for prescribing disciplined program design (program structuring and data structuring) and implementation. Architecture and partitioning rules, documentation conventions, configuration and data management procedures, etc. are among those standards to be disseminated. A standards enforcer is a computer program used to automatically determine whether prescribed programming standards and practices have been followed. The program can check for violations to standards set for such conventions as program size, commentary, structure, etc.

60. Statement-Level Simulator. A computer program used to simulate the execution characteristics of a target computer at the source instruction-level using a sequence of instructions of a host computer. The execution characteristics are approximately equal to those of the target computer for which the source code is ultimately intended.

61. Static Analysis. A technique used during test to identify weaknesses in the source code. The syntax of a program is examined and statistics about it are generated. Items such as relationships between module, program structure, error prone constructions, and symbol/subroutine cross-references are checked.

62. Static Analyzer. A computer program used to provide information about the features of a source program. This type of tool examines the source code statically (not under execution conditions) and performs syntax analysis, structure checks, module interface checks, event sequence analysis, and other like functions. The National Bureau of Standards Static Fortran Analyzer serves as one example. It samples Fortran programs and collects statistics on the utilization of predetermined syntactic constructs.

63. Stress Testing. A technique employed to confirm that the code performs its specifications satisfactorily under extreme operating conditions computing nominally correct values for worst case input values (i.e., singularities, end points for the range of data, etc.).

64. Structured Programming. The technique used to develop structured programs (i.e., use a limited number of logic structures). Associated with the technique are certain practices such as indentation of source code to represent nested logic levels, the use of meaningful data names, and descriptive commentary. Numerous reports are available on this subject.

65. Symbolic Execution. A technique that employs symbolic data to confirm that the software performs properly. Symbolic execution allows one to choose intermediate points in the test spectrum ranging from individual test runs to correctness proofs. Its results can be used to develop a minimum set of test cases.

66. System Simulations. Computer system simulation is a technique used to predict system performance by exercising a model of the system hardware/software over time. Simulation based on well-planned experiments representative of the real-world environment will produce results that help verify and improve system performance. The simulations are also used to help predict how the system will react to alternative loads with modified configurations. Specific language systems such as ECSS, CSS, SCERT, and SAM have been devised as aids to implementation.

67. Test Beds. A test site comprised of actual hardware (hardware test site) or simulated equipment (software test site). A hardware test site uses the actual computer and interface hardware to checkout the hardware/software interfaces and actual input/output. The program execution is conformed using actual hardware timing characteristics, but the output is limited. A software test site uses an instruction-level and/or statement-level simulator to model actual hardware. A software test site permits full control of inputs and computer characteristics, allows processing of intermediate outputs without destroying simulated time and allows full test repeatability and diagnostics.

68. Test Drivers, Scripts, Generators. To run tests in a controlled manner, it is often necessary to work within the framework of a "scenario" - a description of a dynamic situation. To accomplish this, the input data files for the system must be loaded with data values representing the test situation or events to yield recorded data to evaluate against expected results. These aids permit generation of data in external form to be entered automatically into the system at the proper time.

69. Test-Result Processor. A computer program used to perform test output data reduction, formatting, and printing. Some perform statistical analysis where the original data may be the output of a monitor.

70. Timing Analyzer. A computer program that monitors and prints execution time of all program elements (functions, routines, and subroutines).

71. Top-Down Programming. The concept of performing in a hierarchical sequence detailed design, code, integration, and test. Numerous reports are available describing this subject.

72. Trace. A computer program that records the chronological sequence of events taken by a target program during its execution.

73. Units Consistency Analyzer. A computer program that analyzes source code to verify units consistency for each usage of a given parameter.

74. Walkthrus. A technique used for reviewing the design or code and identifying discrepancies. The responsible analyst discusses his product with his peers and solicits their constructive advice. Product modifications are then made at the discretion of the analyst to correct problems identified during the review.

75. Workload Evaluation Aids. Computer programs written to evaluate whether or not a specific computer or computer configuration (both hardware and software) can handle the projected workload expressed as a series of applications (benchmark), a projected series of applications expressed as an instruction mix (synthetic program), or a set of basic user functions (kernal).

APPENDIX C

BIBLIOGRAPHY

1. Asch, A., D. W. Kelliher, J. P. Locher and T. Connors, DOD Weapon Systems Software Acquisition and Management Study, MITRE Corporation Report No. 6908, June 1975.

2. Augrist, E. F., A Survey of Avionics Simulation Facilities, FEDSIM Report, MV-409-012-TAC/AFDAA, August 1974.

3. Bartlett, J. C., Software Validation Study, Logicon Report DS-72210-R13, March 1973.

4. Basili, V. R. and R. E. Noonan, "A Testing Tool for a Fire-Control Environment," COMPCON Fall, September 1976.

5. Biche, P. W., D. J. Morgan and R. E. Wattenberg, Independent Test and Evaluation Guidelines, Logicon Report No. DS-R74036, 26 July 1974.

6. Boehm, B. W., "Software and Its Impact: A Quantitative Assessment," Datamation, May 1973.

7. Boehm, B. W., "The High Cost of Software," Practical Strategies for Developing Large Software Systems, Addison-Wesley, 1975.

8. Bratman, H. and M. C. Finfer, Software Acquisition Management Guidebook: Verification, ESD-TR-77-263, August 1977.

9. Brosius, D. B., Software Validation Study, Autonetics Report C73-11/201, February 1973.

10. Brown, J. R., "Why Tools," Computer Science and Statistics: 8th Annual Symposium on the Interface, University of California at Los Angeles, February 1975.

11. Buckley, F. J., "Verification of Software Programs," Computers and Automation, February 1971.

12. Buckley, F. J., "Software Testing - A Report From the Field," IEEE Symposium on Computer Software Reliability, May 1973.

13. Chandler, A. R., "Software Verification and Validation for Command and Control Systems," RCA Engineer, Vol. 19, No. 5, 1974.

14. Clarkson, W.K., Computer Program Validation/Verification, Aerospace Corporation Report No. TOR-0059(9320)-16, Februry 1971.

15. Culpepper, L.M., "A System for Reliable Engineering Software," International Conference on Reliable Software, April 1975.

16. Curry, R.W., J.S. Hickey and F.P. Konieczny, Reliable Software, Science Applications Incorporated Report No. SAI-76-598-HU, 5 November 1975.

17. DeRoze, B.C., "An Introspective Analysis of DOD Weapon System Software Management," Defense Management Journal, October 1975.

18. DeWolf, J.B. and J. Wexler, "Approaches to Software Verification with Emphasis on Real-Time Applications," AIAA Computers in Aerospace Conference, Los Angeles, California, October 31 - November 2, 1977.

19. Dreyfus, J.M., Use of Simulation in the BMD Systems Technology Program Software Development, TRW Report No. TRW SS-76-10, October 1976

20. Drezner, S.M., H. Shulman and W.H. Ware, The Computer Resource Management Study, RAND Report R1855-PR, September 1975.

21. Erliane, R.V., et al, Verification of the Titan IIIC-26 Operational Flight Program, Aerospace Corporation Report No. TOR-0074 (4411)-4, January 1974.

22. Fairley, R.E., "An Experimental Program-Testing Facility," IEEE Transactions on Software Engineering, Volume SE-1, Number 4, December 1975.

23. Felty, J.L. Software Support Tools, Intermetrics Report No. IR-204-2, 15 October 1976.

24. Furtaw, R.N., Aircraft Avionics Tradeoff Study, ASD/XR73-19, November 1973.

25. Gibson, C.G. and J.R. Railing, Verification Guidelines, TRW Report No. SS-71-04, TRW Software Series, August 1971.

26. Green, E. "What, How and When to Test," Workshop on Currently Available Test Tools - Technology and Experience, April 1975.

27. Griffith, W.V., Aircraft Avionics Tradeoff Study, ASD/XR73-20, November 1973.

28. Gruenberger, F., "Program Testing and Validation," Datamation, July 1968.

29. Hartwick, R.D., The Advanced Targeting Study, SAMSO-TR-71-124, June 1971.

30. Herring, F.P. and C.J. Mabee, Survey of Support Software for Operational Flight Programs and Avionics Integration Support Facility Software, TRW Report No. 28675-6232-RU-00, May 1977.

31. Hetzel, W., Program Test Methods, Prentice-Hall, Englewood Cliffs, N.J., 1973.

32. Howley, P.P. and R.W. Scholten, "Test Tool Implementation" AIAA Computers in Aerospace Conference, Los Angeles, California, October 31 - November 2, 1977.

33. Kennedy, J.E., A Survey of Automated Computer Program Verification Tools, Aerospace Corporation Report No. TOR-0075 (5112)-1, 15 August 1974.

34. Kessler, M.M. and W.E. Kister, Software Tool Impact, RADC TR-74-300, Structured Programming Series, Volume XIV May 1974

35. Kosy, D.W., Air Force Command and Control Information Processing in the 1980's: Trends in Software Technology, RAND, June 1974 (AD 525 661L).

36. Lane, R.J., The N² Chart, TRW Report No. TRW-SS-77-04, November 1977.

37. Logicon Inc., Verification and Validation for Terminal Defense Program Software, Report No. HR-74012, 31 May 1974.

38. Manley, J.H., "Embedded Computer System Software Reliability," Defense Management Journal, October 1975.

39. Martin Marietta Corporation. Software Verification and Validation Report, Operational Flight and Ground Program, Report No. MCR-73-233, December 1973.

40. McClean, R.K. and B. Press, The Flexible Analysis, Simulation, and Test Facility: Diagnostic Emulation, TRW Report No. TRW-SS-75-03, October 1975.

41. Miller, E.F., A Survey of Major Techniques of Program Validation, General Research Corporation Report No. RM-1731, October 1972

42. Miller, E.F., Methodology for Comprehensive Software Testing, RADC-TR-75-161, June 1975.

43. Miller, E.F., "Program Testing: Art Meets Theory," Computer, July, 1977.

44. Moranda, P., G. Foshee, M. Kirchoff and R. Omre, Final Report, A Methodology for Producing Reliable Software, McDonnell Douglas Astronautics Company - West Report No. MDC G7210, Vol. I and II, March 1976.

45. Mullin, F.J., "Considerations for a Successful Software Test Program," AIAA Computers in Aerospace Conference, Los Angeles, California, October 31 - November 2, 1977.

46. Osterweil, L.J. and L.D. Fosdick, Data Flow Analysis as an Aid in Documentation, Assertion Generation, Validation and Error Detection, Department of Computer Science, University of Colorado, September 1974.

47. Osterweil, L.J., "A Methodology for Testing Computer Programs," AIAA Computers in Aerospace Conference, Los Angeles, California, October 31 - November 2, 1977.

48. Paige, M.R. and E.F. Miller, Methodology for Software Validation - A Survey of the Literature, General Research Corporation Report No. RM-1549, March 1972.

49. Pomeroy, J.W., "A Guide to Programming Tools and Techniques," IBM Systems Journal, Vol. II, No. 3, 1972.

50. Ramamoorthy, C.V. and S.B.F. Ho, "Testing Large Software with Automated Software Evaluation Systems," IEEE Transactions on Software Engineering, Vol. SE-1, March 1975.

51. Reifer, D.J., Computer Program Verification/Va[illegible]lation/Certification, Aerospace Corporation Report No. TOR-0074(4112,[illegible]), 30 May 1974.

52. Reifer, D.J. and R.L. Ettenger, Test Tools: Are They a Cure-all?, SAMSO-TR-75-13, 15 October 1974.

53. Reifer, D.J., "Automated Aids for Reliable Software," International Conference on Reliable Software, April 1975.

54. Reifer, D.J., Interim Report on the Aids Inventory Project, SAMSO-TR-75-184, 16 July 1975.

55. Reifer, D.J., A New Assurance Technology for Computer Software, SAMSO-TR-75-238, 1 September 1975.

56. Reifer, D.J., Microprogram Verification and Validation, SAMSO-TR-76-217, February 1976.

57. Reifer, D.J. and F. Dyke, "Computer Program Verification and Validation: A SAMSO Perspective," Conference on Managing the Development of Weapon System Software, May 1976.

58. Reifer, D.J., "Computer Program Verification and Validation," Invitational DOD/Industry Conference on Software Verification and Validation, August 1976.

59. Reifer, D.J. and S. Trattner, "A Glossary of Software Tools and Techniques," Computer, Vol. 10, No. 7, July 1977.

60. Reifer, D.J., "The Software Engineering Checklist," AIAA Computers in Aerospace Conference, November 1977.

61. Reifer, D.J., "Software Quality Assurance Tools and Techniques," Software Quality Management, Petrocelli, to appear 1979.

62. Rubey, R.J., "Quantitative Aspects of Software Validation," International Conference on Reliable Software, April 1975.

63. Smith R.L., Validation and Verification Study, RADC TR-74-300, Structured Programming Series, Vol. XV, May 1975.

64. Straeter, T., "Research Flight Software Development," NASA/ICASE University Conference on Flight Software Development, 11 January 1977.

65. Stucki, L.G., Software Automated Verification System Study, McDonnell Douglas Astronautics Company - West Report No. MDC-G5103, January 1974.

66. Stucki, L.G., "Tools - Lessons Learned - New Strategies," Computer Science and Statistics: 8th Annual Symposium on the Interface, University of California at Los Angeles, February 1975.

67. Thayer, R.H. and E.S. Hinton, "Software Reliability: A Method That Works," Proceedings of the National Computer Conference, AFIPS, 1975.

*U.S.Government Printing Office: 1980 — 687-084/29